Commodore WORLD

Nº 3. ABRIL 1984

275 PTAS.



DESCRIPCIÓN ALFABÉTICA DE LOS MNEMÓNICOS DEL 6502/6510 (III)

BVS

Salta si hay desbordamiento

Operación: Salta si V = 1

(Ref.: 4.1.1.7)

N Z C I D V
\- - - - - -

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Relativo	BVS Oper.	70	2	2*

* Suma 1 si se salta a la misma página.
* Suma 2 si se salta a otra página.

CLC

Borra la bandera de acarreo

Operación: 0 - >C

(Ref.: 3.0.2)

N Z C I D V
\- - 0 - - -

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	CLC	18	1	2

CLD

Borra el modo decimal

Operación: 0 → D

(Ref.: 3.3.2)

N Z C I D V
\- - - - 0 -

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	CLD	D8	1	2

CLI

Borra el bit de desactivación de interrupción

Operación: 0 → I

(Ref.: 3.2.2)

N Z C I D V
\- - - 0 - -

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	CLI	58	1	2

CLV

Borra la bandera de desbordamiento

Operación: 0 → V

(Ref.: 3.6.1)

N Z C I D V
\- - - - - 0

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	CLV	B8	1	2



SUMARIO 3

Abril 1984

CONTENIDO — PAG.

CONTENIDO	PAG.
EDITORIAL	4
CLAVE PARA INTERPRETAR LOS LISTADOS	6
MAS ALLA DEL MANUAL Este mes aprenderemos a dominar los bucles GOTO y los comandos POKE	7
ATAJO HACIA EL COLOR Sin el manual puede resultar muy pesado recordar los cógidos de color	10
COMO EXPRIMIR EL JUGO AL LAS TECLAS DE FUNCIONES Este programa te permite hacerlo con facilidad	12
VENTANA CBM Continuamos con algunos aspectos del sistema co-residente MEC/DOS	18
VIDEOCASINO Mosqueando al mosquito	24
CLUB COMMODORE	28
Golf	29
Las siete y media	29
Gráficos en tres dimensiones	31
Programas Sala y Diccionario	33
Comando Old	38
El Misterio de los Pokes	39
Glosario	39
Rincón del 700	40
Prints sofisticados	42
Galería de Soft (novedades)	44
Magia	46
De tomo y lomo	48
Carta Blanca	49
Marketclub	52
Las aventuras de Ramy	54
EL MANEJO DE FICHEROS Solucionando problemas con la base de datos "Deluxe"	58
EXCURSION POR LA MICROPROGRAMACION Cursillo de lenguaje máquina	62
CARACTERES ESPECIALES CON EASY SCRIPT Cómo generar caracteres especiales	64
METEDURAS DE PATA Y CORRECCIONES DEL MANUAL	66

PROXIMO NUMERO

- Dibujando con los Commodores
- Romance con el VIC
- Animaté con los Sprites
- Crear un Vicasso
- Ahorrando Memoria en el VIC
- Los Arrays a raya
- La Caja de los Misterios
- Programa de Evaluación para colegios
- Juegos
- Club Commodore

Y Todas vuestras colaboraciones

¡OS ESPERAMOS!

Commodore World es publicado en colaboración entre Microelectrónica y Control-Commodore y SIMSA

EQUIPO

Manuel AMADO; José Luis ERRAZQUIN; Adela LOPEZ; María LOPEZ; Juan MARTINEZ; Pere MASATS; Jeffrey MILLS; Rafael NAVARRO; Fernando M. RODRIGUEZ; Diego ROMERO; Albert SANGLAS; Manuel SANS; Jordi SASTRE; Valerie SHANKS...

... Y NUESTROS LECTORES

SIMSA

Coordinador María López

Pedro Muguruza, 4-8ºB — Madrid-16 — Tlf.: (91) 259 54 78

Sant Gervasi de Cassoles, 39 despacho 4

Barcelona-22 — Tlf.: (93) 212 73 45

MICROELECTRONICA

Y CONTROL-COMMODORE

Coordinador Pere Masats

Taquigrafo Serra, 7-5º

Barcelona 29

Teléfono (93) 250 51 03/02

Imprime: IBERDOS, S.A. Germán Pérez Carrasco, 24. Madrid-27

Depósito Legal: M-2944-1984

EDITORIAL

SUPER-INTERESANTISIMO

... Viaje a Estados Unidos e Inglaterra...

... YA FALTA POCO...

Ya falta poco... hoy a 20 de marzo hemos anotado el socio suscriptor 4154; nuestro socio-suscriptor 5000, como ya decíamos en nuestro número anterior, le corresponderá un viaje a Estados Unidos con visita a las instalaciones de la Casa Commodore y entre los 4999 suscriptores primeros se sorteará un viaje a Inglaterra con visita, asimismo, a la central de Commodore allí.

Número de socio suscriptor

A partir del próximo número en la etiqueta de envío de la revista irá el número de socio-suscriptor.

COLABORACIONES

Es estupendo ver la gran respuesta del equipo de lectores. Si no veis vuestra colaboración publicada en el primer número después de enviada no os defraudéis, tened paciencia —las colaboraciones que nos llegan después del día 10 de cada mes deben quedar para el número siguiente por razones de maquetación e imprenta.

IMPORTANTE: Repetimos que los programas procurad enviárnoslos en cinta pues nos hace mucho más fácil el trabajo de comprobación —devolvemos siempre las cintas con un programa de regalo. Si os es imposible enviar cinta por alguna razón de peso... bueno... que le vamos a hacer!— pero sólo por una razón de gran peso, por favor. Si tenéis teléfono, enviad el número con las señas por si necesitamos pedir alguna aclaración.

IMPORTANTE OTRA VEZ: Aunque comprobamos los programas de colaboración, no nos hacemos responsables de los mismos.

SERVICIO DE CINTAS

Como ya hemos dicho en números anteriores, aquellos lectores que quieran las cintas de los programas que aparecen en la revista pueden pedirlos enviando un cheque de 875 pesetas por cinta más 75 pesetas por gastos de envío pero —CUIDADO— **no podemos enviar cintas con programación de los colaboradores**, para esto tenéis sus nombres y direcciones a fin de que podáis poneros en contacto directamente.

CALIDAD DE IMPRESION DE LOS PROGRAMAS

Un par de lectores se han quejado de la calidad de impresión de los listados de los programas. Como ya hemos dicho, todos los programas que se publican van comprobados al detalle. Los compositores de imprenta, que no son expertos en informática, pueden cometer errores y en un listado es casi imposible evitar a ese duendecillo travieso de imprenta. Por esta razón, y para evitar errores frustantes para todos, entregamos los programas a la imprenta para reproducir en directo del listado original confeccionado por nuestra impresora. Creemos que es preferible sacrificar un poco la calidad de impresión a la calidad auténtica que pretendemos dar. De todas formas, a partir de este número se reproducen los programas a un tamaño "adecuado" para la vista.

MAGIA Y CONCURSOS

Contestando la duda de un lector, y por si hubiera otros con esta misma duda, queremos aclarar que cuando se haga el sorteo de premios entre los trucos aparecidos en Magia, solamente entrarán en sorteo aquellos trucos que nos hayan sido remitidos directamente por nuestros lectores, sean suscriptores o no. Aquellos trucos que hayan sido utilizados de nuestra revista americana no entrarán en ningún caso en sorteo.

Siempre que nuestra revista COMMODORE WORLD convoque un concurso de cualquier tipo, publicará el nombre completo de los ganadores con dirección, edad y número de socio en caso de ser sucriptor.

FERIA DEL MICRO EN EL CORTE INGLES

A partir del día 2 de Abril, y durante todo el mes, el Corte Inglés de Castellana, en Madrid, celebrará su primera Feria del Micro —allí estará Commodore y allí estaremos nosotros con Commodore World, esperamos que madrileños y "turistas" vengáis a visitarnos— no os olvidéis, todo el mes de abril en la planta 6ª del Corte Inglés de Castellana. ¡Y con buena cafetería allí mismo para tomarse un bocata!

VIAJES A ESTADOS UNIDOS E INGLATERRA PARA NUESTROS SUSCRIPTORES

¡CUIDADO!
Solamente la revista COMMODORE WORLD publicada por S.I.M.S.A. está ofreciendo esta oportunidad a sus socios-suscriptores

TIRADA DE COMMODORE WORLD
Certificación de 28.000 ejemplares

Según certificado firmado por la auditoría internacional
DELOITTE HASKINS SELLS, S.A.
el 5 de marzo de 1984,
la tirada de Commodore World es de 28.000 ejemplares.
Este certificado
se encuentra en las oficinas de S.I.M.S.A.
para comprobación de cualquier persona o entidad que lo desee.

SERVICIO DE CINTAS

Título del programa .... publicado en nº ☐
Título del programa .... publicado en nº ☐
Título del programa .... publicado en nº ☐
Título del programa .... publicado en nº ☐

Precio por cinta 850 pesetas. Gastos de envío 75 pesetas. Forma de pago por cheque.

Peticionario ....
Calle .... Nº
Población .... D.P. .... Provincia ....
Incluyo cheque por .... pesetas.

EJEMPLARES ATRASADOS DE «CLUB COMMODORE»

Para poder satisfacer la creciente demanda de números atrasados de nuestra Revista, agotada en todas sus ediciones, hemos puesto en marcha un Servicio para suministrar fotocopias de los ejemplares que nos sean solicitados. Para recibir las fotocopias de una o de varias ediciones, no hay más que enviarnos el boletín con los datos indicados.

SERVICIO DE FOTOCOPIAS.— NUMERO DE LA EDICION SOLICITADA.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(Poner una X debajo del número de edición pedido)

Peticionario: D. ....
Calle .... nº ....
Población .... D.P. .... Provincia ....

Forma de pago por cheque **Precio de la edición fotocopiada: 250 ptas.**
La colección completa del 0 al 15: 2.500 ptas. + 100 ptas. por gastos de envío
Incluyo cheque por .... ptas.

Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)

[CRSRU] = Tecla cursor hacia arriba (con SHIFT)

[CRSRR] = Tecla cursor a la derecha (sin SHIFT)

[CRSRL] = Tecla cursor a la izquierda (con SHIFT)

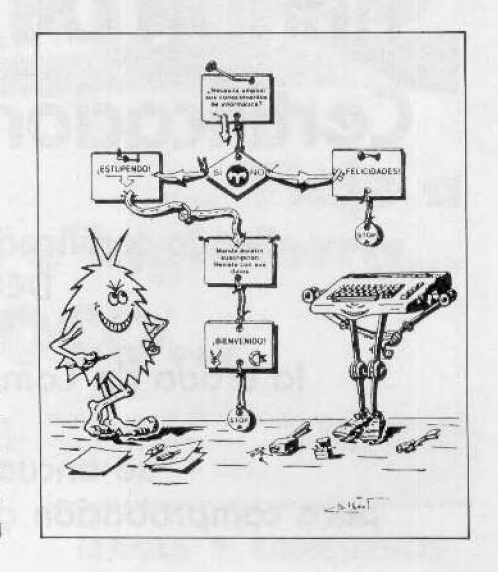
[HOME] = Tecla CLR/HOME (sin SHIFT)

[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. ■



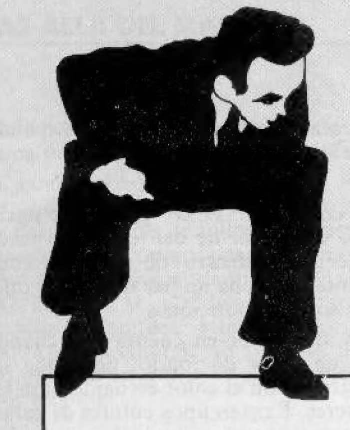
INDICE ANUNCIANTES

	Págs.
Basic Microordenadores, S. A.	43
Commodore 64	36 y 37
Commodore World	51
EAF	17
El Corte Inglés	26 y 27
Iberdata	57
MicroSistemas	23 y 53
Sakati	33
Tecnhel	31
Tele Sant Just	65
Tronic	41
VIC-20	68



Pedro Muguruza, 4-8º B
Teléf.: 259 54 78
MADRID-16

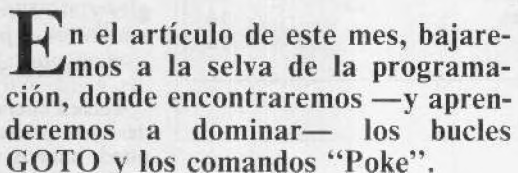
Sant Gervasi de Cassoles, 39-despacho 4
Teléf.: 212 73 45
BARCELONA-22



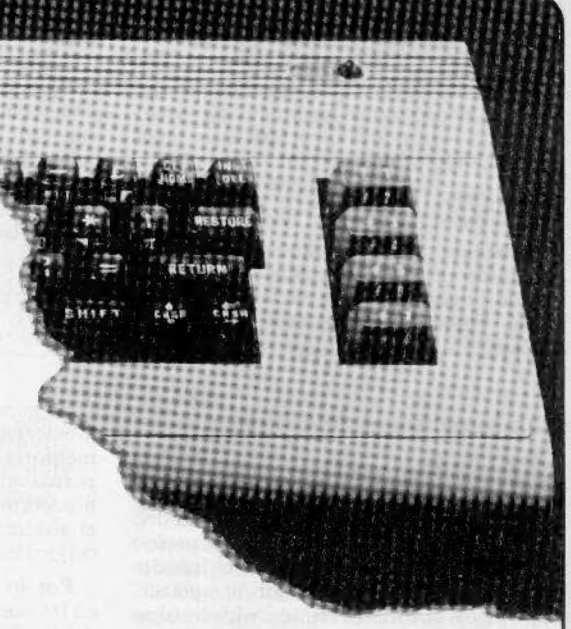
Una Excursión en Basic
Más allá del Manual

Capítulo III

Por Jeffrey Mills



En el artículo de este mes, bajaremos a la selva de la programación, donde encontraremos —y aprenderemos a dominar— los bucles GOTO y los comandos "Poke".



Hasta ahora en esta serie de artículos, hemos hablado del uso de varios comandos internos, el uso de las variables, el comando Read y la sentencia Data, y nos hemos aproximado a los bucles GOTO. (El Listado 1 presenta el estado actual del programa que estamos desarrollando).

Este mes vamos a hablar un poco más del bucle GOTO. En general, el concepto de los bucles constituye un método para repetir la misma cosa una y otra vez. Por ejemplo, si yo te digo que abras cinco bolsas de papel y que mires su contenido, harás la misma operación para cada bolsa.

Resultaría un poco pesado tener que escribir cinco veces la explicación de este proceso, pero si la instrucción se coloca dentro de un bucle de control, sólo hace falta escribirla una vez. El bucle ejecutará la instrucción cinco veces.

Para controlar un bucle utilizando una sentencia GOTO, también hace falta verificar el número de veces que se va a realizar la instrucción. Se puede utilizar la sentencia If...Then (pág. 37-39 en el manual del C-64) para comprobar alguna relación específica entre dos o más datos. Los datos pueden ser variables, constantes (el nombre que damos a los datos no variables) o las dos cosas.

En el caso de las cinco bolsas de pepel, éstas se van contando a medida que vayan llegando y se dejan de contar cuando se hayan comprobado las cinco. Otro método para realizar la verificación sería determinar el momento en que no queden bolsas por mirar.

Cómo salir del Bucle

Para traducir esto en el programa Catalog, cuando se hayan imprimido todos los datos de la sentencia Data, se deja de imprimir. Esto se puede hacer colocando unos datos "falsos" al final de las sentencias Data. Sería poco probable que una cinta tuviera el número -1, y que un juego se llamara EOF (Fin de Fichero). Ahora se añade la falsa sentencia Data tecleando

```
9999 DATA -1, EOF
```

Para realizar la verificación de programa para estos datos se inserta la frase "Si el número de cinta = -1, deja de imprimir". Esto traducido en Basic sería: If N = -1 then (salir del bucle).

Para conseguir una programación bien estructurada, es buena idea comprobar las condiciones de control del bucle desde el principio del conjunto de instrucciones contenidas en un bucle. Esto se hace añadiendo la comprobación en la línea 65. (El bucle empieza en la sentencia Read en la línea 60). Se teclea:

```
65 IF N = -1 THEN GOTO 90
```

Esta instrucción comprueba si el programa ha llegado a los datos falsos. Si es así, el programa pasa a la línea 90; si no, pasa directamente a la siguiente línea, la 70. Cuando llega a la línea 80, vuelve a la línea 60, donde lee el siguiente dato de la sentencia Data y vuelve a comprobar para ver si ha llegado el momento de salir del bucle. El programa sigue dentro del bucle hasta que se encuentren los datos falsos.

Dado que el programa tiene instrucciones de continuar en la línea 90, dicha línea tiene que ser añadida para que el programa tenga a dónde ir.

Si no existe la línea de destino del GOTO, el programa terminará en error. Es buena idea poner un REM en la línea 90 indicando el final del bucle.

Se teclea:

90 REM***FIN DE BUCLE***

A medida que vayas adquiriendo conocimientos de programación, descubrirás muchos más usos para la sentencia If... Then. Hablaremos de estos usos más adelante.

Combinaciones de Color

Vamos a volver a hablar de la sentencia Print para ver cómo la salida del Commodore puede resultar más interesante. Podría ser divertido que el listado salga en color, y a lo mejor te apetece cambiar los colores del fondo y los bordes de la pantalla.

Todos los controles para que los datos salgan en pantalla están codificados de modo permanente en forma numérica dentro de la memoria del ordenador. Parte de esa memoria está disponible para el uso, así que el usuario puede modificar la forma en que funciona el ordenador. Esto se hace mediante el comando "Poke".

Tabla 1. Valores de POKE para los colores del C-64.

COLOR	VALOR
Negro	**0**
Blanco	**1**
Rojo	**2**
Cian	**3**
Púrpura	**4**
Verde	**5**
Azul	**6**
Amarillo	**7**
Naranja	**8**
Marrón	**9**
Rosa	**10**
Gris 1	**11**
Gris 2	**12**
Verde claro	**13**
Azul claro	**14**
Gris 3	**15**

El comando "Poke" modifica el valor almacenado en una posición específica de memoria. Pero no te preocupes; no es permanente. Se puede volver a modificar mediante otro comando "Poke". Además, el sistema vuelve a fijarse en los valores originales cuando se apaga el ordenador.

Por lo tanto, los valores almacenados en las posiciones que controlan los colores del fondo y los bordes de la pantalla pueden ser modificados.

La posición 53280 controla el color de los bordes y la posición 53281 controla el color del fondo de la pantalla. La Tabla 1 presenta los valores "Poke" para dichas posiciones para los diversos colores.

Estos valores pueden ser colocados en cualquiera de las dos posiciones de memoria que controlan los colores de pantalla.

El formato correcto para un comando "Poke" es:

POKE 53280,7

Un comando "Poke" puede ser introducido directamente del teclado o puede ser ejecutado dentro de un programa. Adelante... prueba un par de valores diferentes en estas posiciones.

Hay que tener en cuenta que cuando eliges un color de fondo, tiene que ser compatible con el color escogido para los caracteres. Existen unos colores de caracteres que no se ven muy bien contra ciertos colores de fondo.

Para el programa Catalog, vamos a poner un color suave de pantalla. Un borde gris alrededor de un fondo negro podría dar buen resultado, pero puedes utilizar el color que quieras. Para emplear el gris y el negro, se añaden las siguientes sentencias al principio del programa:

25 POKE 53280,12: POKE 53281,0

Ten en cuenta que se puede incluir más de una instrucción en una línea. Esto se puede hacer si se separan las instrucciones mediante dos puntos (:). Es importante tener en cuenta que si REM es la primera "palabra" en una línea, dicha línea no se ejecuta. Si quieres utilizar REM en una línea de varias sentencias, se debe de emplear como la **última** sentencia en la línea.

En el manual del usuario no hay una sección específica que trate de los colores ni de los listados en color. Tendrás que comprobar el índice para saber cuáles son las páginas que se refieren a este tema.

Comandos Anidados

En el primer artículo de esta serie, hice una breve referencia a los comandos anidados que se pueden añadir

LISTADO 1. EL PROGRAMA CATALOG EN SU ESTADO ACTUAL

```
10 REM*** PROGRAMA CATALOGO DE CINTA
 ***
20 REM*** ESCRITO POR : TU NOMBRE
 ***
30 PRINT"[CLR]"
40 PRINT"[9SPC]CATALOGO[SPC]DE[SPC]C
INTA"
50 PRINT
52 PRINT"NO.","NOMBRE"
54 PRINT"CINTA","PROGRAMA"
56 PRINT
60 READN,P$
70 PRINTN,P$
80 GOTO60
9000 DATA101,JUEGO1,101,JUEGO2
9010 DATA102,JUEGO3,102,JUEGO4
9020 DATA103,JUEGO5,103,JUEGO6
```

LISTADO 2. PROGRAMA QUE DEMUESTRA LOS COMANDOS ANIDADOS DE COLOR

```
10 REM*** PROGRAMA CATALOGO DE CINTA
 ***
20 REM*** ESCRITO POR : TU NOMBRE
 ***
25 POKE53280,12:POKE53281,0:REM:REM
*** COLORES DE PANTALLA ***
30 PRINT"[CLR]"
40 PRINT"[YEL][9SPC]CATALOGO[SPC]DE[SPC]
CINTA"
50 PRINT
52 PRINT"[COMM6]NO.","NOMBRE"
54 PRINT"CINTA","PROGRAMA"
56 PRINT
60 READN,P$
65 IFN=-1THENGOTO90
70 PRINT"[CYN]"N,P$
80 GOTO60
90 REM*** FIN DEL BUCLE ***
9000 DATA101,JUEGO1,101,JUEGO2
9010 DATA102,JUEGO3,102,JUEGO4
9020 DATA103,JUEGO5,103,JUEGO6
9999 DATA-1,FIN DE ARCHIVO
```

a las sentencias Print dentro de un programa para obtener diversos resultados.

Cuando se teclea una sentencia Print, ciertas combinaciones de teclas introducidas después de las comillas iniciales funcionarán como códigos para determinar el color de lo que sigue (dentro de las comillas) cuando se ejecuta el programa. Cuando se emplea este método para modificar los colores, un símbolo invertido aparece en pantalla para indicar el color elegido.

Si estos códigos de color se teclean en sentencias directas (**no** anidados dentro de las comillas), los símbolos no aparecerán; lo que ocurre es que se modifica el color de lo que se teclee a continuación. La Tabla 2 presenta las combinaciones de teclas para los distintos colores, junto con los símbolos indicativos que aparecen si el comando está anidado dentro de comillas.

Ahora intenta que salga el título del listado del catálogo en amarillo. Lista el programa, desplaza el cursor a la línea 40 y colócalo justo detrás de las primeras comillas en dicha sentencia. A continuación pulsa las teclas "shift" y INST/DEL y teclea:

CTRL 8

Pulsa la tecla "Return". Debe de aparecer un π invertido como el primer carácter que será imprimido. (Ver el Listado 2).

Tabla 2. Las combinaciones de teclas y los símbolos para los colores en el C-64.

TECLAS PULSADAS	COLOR	CARACTER INVERTIDO EN PANTALLA
CTRL 1	Negro	Shift-P
CTRL 2	Blanco	E
CTRL 3	Rojo	Libra
CTRL 4	Cian	Shift-*
CTRL 5	Púrpura	Logo- -
CTRL 6	Verde	Flecha Arriba
CTRL 7	Azul	Flecha Abajo
CTRL 8	Amarillo	π
LOGO 1	Naranja	Pico
LOGO 2	Marrón	Shift-U
LOGO 3	Rosa	Shift-V
LOGO 4	Gris 1	Shift-W
LOGO 5	Gris 2	Trebol
LOGO 6	Verde claro	Shift- -
LOGO 7	Azul claro	Diamante
LOGO 8	Gris 3	Shift- +

Todo lo que siga a continuación será en amarillo hasta que se cambie a otro color o se pulsan las teclas run/stop y restore.

Ahora vamos a convertir las cabeceras de las columnas en verde claro.

Siguiendo el procedimiento anterior, se modifica la línea 52 para que incluya un LOGO 6 como el primer carácter que será imprimido. Dado que así se modifica el color imprimido a verde claro hasta que se vuelva a modificar, vamos a probar añadiendo otro carácter de color en la línea 70. Esto modificará el color de toda la salida del programa (el mismo listado del Catálogo). ¿Y el color cian?

Esta inserción tendrá que incluir unas comillas iniciales, CTRL 4, y unas comillas finales. La Línea 70 quedará de la siguiente forma:

70 PRINT"(reverse shift-*)"N,P$

No debe de quedar un espacio entre las comillas y el carácter invertido. Si lo hubiera, el espacio también sería imprimido, dejando las columnas imprimidas fuera de alineación.

LISTADO 3. PROGRAMA QUE DEMUESTRA TODAS LAS COMBINACIONES DE COLOR POSIBLES EN EL C-64

```
10 REM*** MUESTRA DE COLOR ***
20 PRINT"[CLR]":REM* BORRA LA PANTAL
LA
30 FORB=0TO15:REM* CADA UNO DE LOS 1
6 COLORES
40 POKE53280,B:REM* ALTERA LA MEMORI
A PARA VISUALIZAR EL COLOR
50 FORS=0TO15:REM* CADA UNO DE LOS 1
6 COLORES
60 POKE53281,S:REM* ALTERA LA MEMORI
A PARA VISUALIZAR EL COLOR
65 PRINT"[CLR]":REM* BORRA PANTALLA
70 FORP=0TO15:REM* CADA UNO DE LOS 1
6 COLORES DE IMPRESION
80 POKE646,P:REM* ALTERA LA MEMORIA
PARA VISUALIZAR EL COLOR
90 PRINT"[HOM][10CRSRD][5CRSRR][RVSON]
[7SPC]COLOR[SPC]BORDE[SPC]="B"[CRSRL]
[10SPC][RVSOFF]"
92 PRINT"[HOM][7CRSRD][5CRSRR][RVSON]
[4SPC]COLOR[SPC]PANTALLA[SPC]="S"[CRSRL]
[10SPC][RVSOFF]"
94 PRINT"[HOM][13CRSRD][5CRSRR][RVSON]
[7SPC]COLOR[SPC]TEXTO[SPC]="P+1"[CRSRL]
[10SPC][RVSOFF]"
100 FORA=1TO250:NEXT:REM* PAUSA PARA
 FRENAR EL PROGRAMA
110 NEXT
120 NEXT
130 NEXT
140 END
```

Te puedes convertir en un mago del color si aprendes los códigos de las teclas en el Commodore 64.

Gama de Colores

Para poder ver todas las combinaciones de colores disponibles en el C-64, teclea un programa corto (Listado 3) después de salvar el programa de Listado Catalog. El programa presentado en el Listado 3 ejecutará una serie de bucles. Sin embargo, estos bucles son For... To ...Next, de los cuales hablaremos en otro artículo.

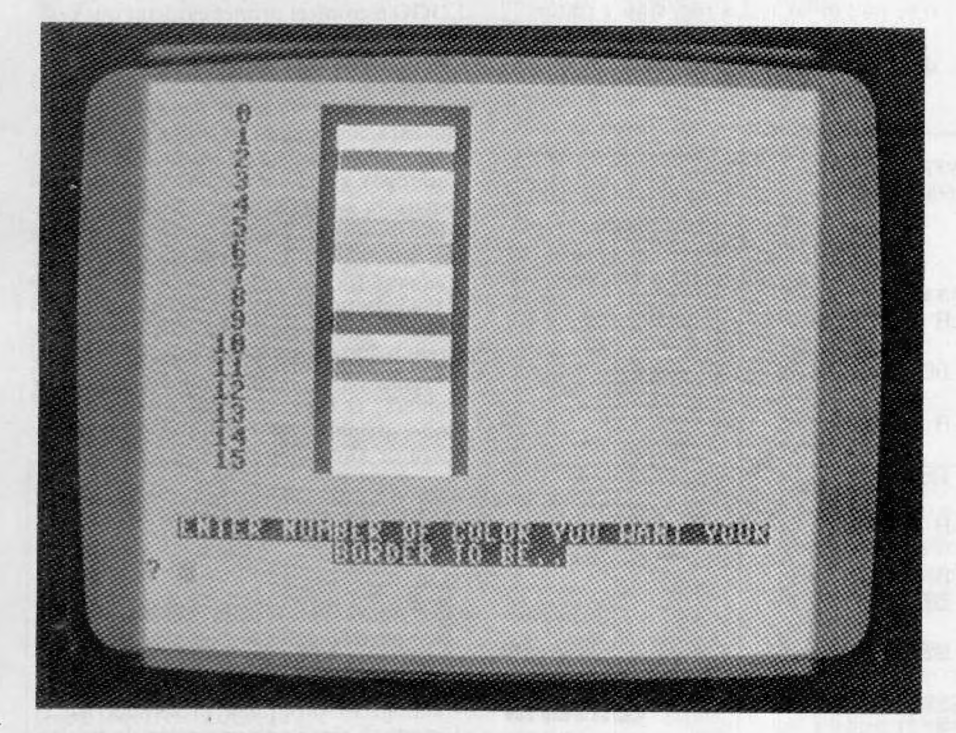
El mes que viene hablaremos del control de la salida del programa, teniendo en cuenta de que ésta puede ocupar más de una pantalla, y de dirigir la salida haciendo que el ordenador haga preguntas.

Atajo hacia el color

Si no tienes el manual a mano, puede resultar muy pesado tener que recordar y utilizar los códigos de color.

POKE 52381,0. Fenomenal; la pantalla se vuelve negra si intento utilizar el color en el Commodore 64. No va muy bien; quizás un gris oscuro funcionase mejor.

Lo único que tengo que hacer es un "Poke" con el código correcto en la posición 53281 y puedo cambiar en seguida el color de fondo de la pantalla. El problema que tengo es que la lista de los códigos de color está en el manual del usuario que se ha quedado en casa.

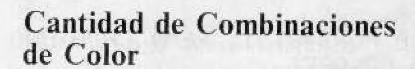


C-64

He llevado el ordenador y la grabadora Datassete a casa de un amigo para enseñarle un programa del cual estoy muy orgulloso. La grabadora me ha permitido cargar el programa, pero estoy utilizando la televisión de mi amigo y el color es horrible. La combinación de color que yo había pensado en casa funcionaba muy bien con mi monitor. Aquí, resulta horrible.

Ahora, desesperado, lo intento con otro "Poke", ¡y la pantalla queda totalmente en blanco! Además, mi mente también lo está y el hecho de que mi amigo se esté riendo a carcajadas no me ayuda en absoluto. Y de repente me doy cuenta de lo que ha pasado; el último "Poke" ha convertido el fondo en el mismo color que el de los caracteres. Si se representan caracteres negros sobre un fondo negro demuestra mucho negro y muy poca inteligencia.

Al final, con unos cuantos "Pokes" salí airoso del problema pero me costó trabajo.



Cantidad de Combinaciones de Color

Existen 16 colores para el borde alrededor de la zona de impresión en la Pantalla, 16 colores para el juego de caracteres y 16 para el fondo. Sin contar con el uso del mismo color para el fondo y el juego de caracteres, existen 16 × 16 × 15, o 3840 diferentes combinaciones de color disponibles en el Commodore 64. Para probarlas todas, tendríamos que hacer muchos "Pokes" en las posiciones específicas de memoria.

La solución más sencilla era la de escribir un programa de utilidad que pudiera formar parte de todos los futuros programas de aplicaciones. Después de todo, una máquina de 64K dispone de suficiente memoria.

Una cosa era cierta; yo no quería tener que hacer el esfuerzo de acordarme de nada. Muchas veces he utilizado programas intensamente durante unas semanas, y luego he pasado a otros proyectos durante unos meses para volver después al programa original. De repente, sería incapaz de recordar que este programa utilizaba el Control W para que toda la salida se desviara a la impresora de 40 columnas. La impresora se quedaría sin hacer nada,

y la única solución sería encontrar la documentación original o revisar el programa todas las veces necesarias para ver cómo funcionaba.

El Programa

El programa "Colocación de Colores" del Listado 1 está diseñado para facilitar el cambio de colores del borde, el fondo y el juego de colores. Todos los 16 colores están presentados al lado de los códigos correspondientes, y el ordenador le pide al usuario que haga su selección. Si el usuario introduce un código que quede fuera de los límites correctos, aparece brevemente un mensaje de error, se vuelven a presentar los colores y se repite la pregunta. El programa también evita que los colores de fondo y los del juego de caracteres sean iguales.

Cuando se introduce el código correcto, entra automáticamente en la posición correcta mediante un "Poke". Por lo tanto, la selección del color se implementa inmediatamente, y el ordenador le pregunta al usuario si resulta satisfactoria. Si la respuesta es afirmativa, se procede con el siguiente parámetro; de no ser así, se repite la operación anterior hasta que el usuario quede satisfecho.

El programa trata los tres parámetros de colores de la misma manera, y al completarse el último, el usuario tiene la oportunidad de repetir todo el proceso. No hace falta que salga el programa hasta que el usuario esté totalmente satisfecho.

El listado del programa está numerado a partir de la línea 9000. Lo diseñé así para usarlo como subrutina para futuros programas, en los cuales se colocaría después del resto del programa. Antes de escribir un programa nuevo, se carga "Colocación de Colores" desde la cinta y se inicia la programación.

En algún punto del programa nuevo, una sentencia GOSUB 9010 le proporciona al usuario la oportunidad de fijar los colores. Evidentemente, la sentencia GOSUB se podría convertir en GOTO si la última línea del programa "Colocación de Colores" se modificara a otra sentencia GOTO en vez de "Return".

Por último, este programa está escrito en Basic y utiliza varias variables numéricas y de "string". Una forma sencilla para evitar que "Colocación de Colores" estropee el programa si dicho programa utiliza las mismas variables (por ejemplo, A$) es llamar "Colocación de Colores" **antes** de que el programa utilice dichas variables.

```
9000 REM FIJAR LOS COLORES
9010 C=1
9020 B$="[SPC]QUE[SPC]NUMERO[SPC]DE[SPC]C
OLOR[SPC]QUIERES[SPC]PARA[SPC]"
9030 C$="EL[SPC]BORDE..?"
9040 D$="EL[SPC]FONDO..?"
9050 E$="LOS[SPC]CARACTERES..?"
9060 REM  VISUALIZAR LAS OPCIONES DE
 COLORES
9070 PRINTCHR$(147):FORA=0TO15
9080 IFA>9THENPRINTTAB(3);A,
9090 IFA<10THENPRINTTAB(4);A;
9100 PRINTTAB(10);CHR$(18);
9110 FORB=0TO8:PRINTCHR$(32);:NEXTB:
PRINT:NEXTA
9120 FORA=0TO15:FORB=1TO7
9130 POKE55306+B+(40*(A+1)),A
9140 NEXTB:NEXTA
9150 PRINT:PRINT:PRINTSPC(2);CHR$(18
);B$
9160 ONCGOTO9170,9180,9190
9170 PRINTSPC(9);CHR$(18);C$:GOTO920
0
9180 PRINTSPC(9);CHR$(18);D$:GOTO920
0
9190 PRINTSPC(6);CHR$(18);E$
9200 INPUTA:IFA>15THENPRINT"DEMASIAD
O[SPC]ALTO":GOTO9230
9210 IFA<0THENPRINT"DEMASIADO[SPC]BA
JO":GOTO9230
9220 GOTO9240
```

```
9230 FORA=1TO1000:NEXTA:GOTO9060
9240 A=INT(A)
9250 ONCGOTO9260,9270,9290
9260 POKE53280,A:GOTO9350
9270 IFPEEK(646)=AGOTO9320
9280 POKE53281,A:GOTO9350
9290 IFA=PEEK(53281)-240GOTO9320
9300 POKE646,A
9310 GOTO9350
9320 PRINT"REPITALO...EL[SPC]FONDO[SPC]Y
[SPC]LOS[SPC]CARACTERES[SPC]NO"
9330 PRINT"PUEDEN[SPC]TENER[SPC]EL[SPC]M
ISMO[SPC]COLOR.":FORA=1TO3000:NEXTA
9340 GOTO9060
9350 PRINT"ESTAS[SPC]SATISFECHO?[SPC](
S/N)"
9360 INPUTA$
9370 IFA$="S"THEN9390
9380 GOTO9060
9390 ONCGOTO9400,9410,9420
9400 C=2:GOTO9060
9410 C=3:GOTO9060
9420 PRINTCHR$(147);"SI[SPC]ESTAS[SPC]S
ATISFECHO[SPC]CON[SPC]LOS[SPC]COLORE
S"
9430 PRINT"QUE[SPC]HAS[SPC]ESCOGIDO,
[SPC]ENTRA[SPC](S)."
9450 INPUTA$:IFA$="S"GOTO9470
9460 GOTO9010
9470 PRINT"TERMINADO"
9480 RETURN
```

Cómo exprimir el jugo a las teclas de funciones

Si has estado buscando un método para definir las teclas de funciones, para satisfacer tus propias necesidades, este programa te permite hacerlo con facilidad.

Cuando compraste tu Commodore, seguramente tenías una duda sobre las teclas de funciones. A lo mejor te quedaste un poco decepcionado la primera vez que pulsaste una de estas teclas, solamente para descubrir que no ocurría nada.

Yo me acuerdo de estar buscando por todo el manual de referencia para saber cómo se asignaban las funciones. Descubrí que las teclas de funciones son muy fáciles de utilizar en los programas en Basic, dado que se pueden introducir, como cualquier otro carácter, pero yo esperaba más. Yo había pensado que podría asignar funciones que me ayudarían con la programación, unas funciones



VIC-20
C-64

que se ejecutarían como si formaran parte del sistema operativo.

Hay un método. Si un "string" predefinido se imprime en pantalla cada vez que se pulsa una tecla de funciones, cualquier función puede ser ejecutada mediante la pulsación de una sola tecla.

Por ejemplo, supongamos que se imprime la palabra LIST cuando se pulsa la tecla F1. Al pulsar la tecla RETURN, se listará el programa. Si la tecla RETURN se hubiese definido como parte del "string", habría sido suficiente pulsar F1 para listar el programa. De modo semejante, mediante la pulsación de la tecla F2 se podría ejecutar un programa.

Yo me doy cuenta, al depurar ciertos programas, que estoy tecleando PRINT PEEK (N), constantemente, cuando N es el número de alguna posición de memoria. Dado que los controles del cursor pueden ser incluidos en los "string", el usuario puede definir un "string" que imprime PRINT PEEK () y devuelve el cursor a la posición justo después del paréntesis de la izquierda. Luego lo único que queda por hacer es teclear el número de la posición que será interrogada, y pulsar la tecla RETURN.

Sin duda, cada usuario tendrá su propia idea a la hora de elegir la función que será asignada a una tecla determinada. Es cuestión de personalizar mi programa a tus propias necesidades y definir las teclas de la manera que más te guste.

Aunque parte de este programa está escrito en lenguaje máquina, no hace falta tener ningún conocimiento de este lenguaje para poder utilizar el programa o definir las teclas de funciones de nuevo. Así que, a seguir leyendo y que las teclas de funciones trabajen para ti.

Usando el programa

Una copia de este programa se presenta en el Listado 1. No es tan largo como aparece, dado que no hace falta teclear las sentencias REM. Es importante salvar una copia del programa antes de que éste se ejecute.

Al teclear el programa, es fácil equivocarse con las sentencias Data, las cuales contienen el programa en lenguaje máquina. Por este motivo, he incluido una prueba al final de cada sentencia Data. El último número de cada sentencia Data es la suma de todos los números anteriores en la misma línea.

Cuando el programa en Basic carga el programa en lenguaje máquina, verifica la prueba en cada sentencia. Si no concuerda, el programa supone que uno, o más, de los números de dicha línea fue mal tecleado, y a continuación se imprime un mensaje de error. El mensaje de error indica exactamente dónde se encuentra la línea incorrecta, lo cual facilita considerablemente la ejecución del programa.

Cuando se ejecuta el programa, un listado debe aparecer en pantalla, demostrando exactamente cómo se definen las teclas de funciones. Después de unos segundos, cuando se haya cargado el programa en lenguaje máquina, el mensaje READY aparecerá en pantalla. El programa en Basic debe de haberse borrado de memoria automáticamente mediante la ejecución de una sentencia New.

En este momento las teclas de funciones

```
0 REM    TECLAS DE FUNCION
1 REM   VERSION COMMODORE 64
2 REM
3 REM     POR JOHN TANZINI
4 REM      Y RON MINDZAK
5 REM
6 REM
7 REM**
8 REM*** LISTA DE VARIABLES
9 REM**
10 REM  T=FINAL DE MAMORIA ACTUAL
15 REM  L= LONGITUD TOTAL DE LOS 'ST
RINGS' DEFINIDOS PARA LAS TECLAS DE
FUNCION
20 REM  SM= PRINCIPIO DEL PROGRAMA E
N CODIGO MAQUINA
25 REM  SS= PRINCIPIO DE LOS 'STRING
S' DEFINIDOS PARA LAS TECLAS DE FUNC
ION
30 REM  PP= PUNTERO A LOS 'STRINGS'
35 REM  SP= PUNTERO A LOS 'STRINGS'
DURANTE SU CARGA
36 REM      UNO DE 'CHECKSUM' (COMPRO
BACION)
40 REM  C= CONTADOR PARA DETERMINAR
SI EL BYTE LEIDO DE UNA ETIQUETA DAT
A ES
45 REM  SU= SUMA DE LOS BYTES. UTILI
ZADO PARA COMPROBAR LOS DATA'S
50 REM  CH= LEXTURA DE SUMA DE LOS D
ATAS
55 REM  B= BYTE LEIDO DE LA SENTENCI
A DATA
96 REM
97 REM**
98 REM*** CALCULAR EL FIN DE MEMORIA
 Y RESERVAR EL ESPACIO PARA LOS 'STR
INGS'
99 REM**
100 T=PEEK(55)+256*PEEK(56)
110 POKE56,PEEK(46)+2
120 POKE52,PEEK(56)
130 PRINT"[CLR]TECLAS[SPC]DE[SPC]FUN
CION":PRINT
196 REM
```

```
197 REM**
198 REM*** DEFINIR LOS 'STRINGS' E I
MPRIMIRLO
199 REM**
200 DIMF$(12)
210 F$(1)="LIST"
220 F$(2)="LOAD"+CHR$(34)
230 F$(3)="RUN"
240 F$(4)="SAVE"+CHR$(34)+"@:"
250 F$(5)="PRINTPEEK([SPC][4SPC])[6CRSRL]"

260 F$(6)="OPEN15,8,15,"+CHR$(34)
270 F$(7)="PRINTFRE(0)"
280 F$(8)="CLOSE15"
290 F$(9)="GOTO"
300 F$(10)="GOSUB"
310 F$(11)="RETURN"
320 F$(12)="POKE"
330 FORI=1TO12
340 PRINT"F"+RIGHT$(STR$(I),LEN(STR$
(I))-1)+"[SPC]=[SPC]";
345 PRINTCHR$(34)+F$(I)+CHR$(34)
350 NEXTI
396 REM
397 REM**
398 REM** DETERMINA LA DIRECCION DE
COMIENZO DE LOS 'STRINGS' Y EL PROGR
AMA
399 REM** DE LENGUAJE MAQUINA
400 L=0
410 FORI=1TO12
420 L=L+LEN(F$(I))
430 NEXTI
435 IFL>231THENPRINT"[RVSON]'STRINGS
 [SPC]DEMASIADO[SPC]LARGOS":GOTO800
440 SM=T-132
450 SS=T-L-156
496 REM
497 REM**
498 REM*** ALMACENAR STRINGS Y PUNTE
ROS
499 REM**
500 PP=SS:SP=SS+12
```

Sigue

están listas para usar. Por ejemplo, si se pulsa la tecla F1, se imprimirá la palabra LIST. Las funciones de F9 a F12 se obtienen mediante la pulsación de la tecla Commodore y una de las teclas de funciones.

Será más fácil iniciar la ejecución del programa si queda bien entendido un punto importante sobre la operación del programa en Basic. Lo primero que hace el programa es trasladar el puntero del final de memoria hacia abajo para proteger un bloque de memoria donde se almacenará el programa en lenguaje máquina.

Si ha habido un error de tecleo en una sentencia Data, el programa lo detectará al calcular la prueba, y bifurcará a la línea 800. En la línea 800 el programa restaurará el puntero del final de memoria, lo cual devuelve toda la memoria al sistema operativo. Si no lo hiciera, el programa no dispondría de la memoria suficiente para ejecutarse correctamente la próxima vez.

Sin embargo, si una sentencia Basic está mal tecleada (provocando un error de sintaxis), el sistema operativo parará el programa en seguida, sin restaurar nada. Si se ejecuta una sentencia GOTO 800 justo cuando se para el programa, no será necesario apagar el equipo, volver a encenderlo y cargar el programa de nuevo. Por supuesto, este problema no surgirá una vez introducido el programa tal y como se presenta en el listado.

Notarás que al pulsar las teclas run/stop y restore, el programa queda desactivado. Es así porque el sistema operativo restaura el vector de interrupciones a su valor original. El programa vuelve a arrancarse mediante la ejecución de un SYS O.

Cuando el usuario ya está seguro de que se ejecuta el programa correctamente, las líneas 611 a 618 pueden ser eliminadas, junto con el último dato de cada sentencia Data. Esta es la parte del programa relacionada con la prueba. El programa en lenguaje máquina tardará la mitad de tiempo en cargarse una vez eliminada la prueba. Es importante eliminar también la última coma de cada sentencia Data.

Dado que el programa en lenguaje máquina permanece en memoria después de que se haya borrado el programa en Basic, se pierde una pequeña cantidad de memoria. La memoria libre queda reducida por 144 bytes, más un byte por cada carácter definido en los "strings".

Volviendo a definir las Teclas de Funciones

Las teclas de funciones F1 a F12 se definen en las líneas 210 a 320. Un "array" de "strings" llamado F$ contiene un "string" relacionado a cada tecla de función. F$(1) es el "string" definido para F1 ; F$(2) es el "string" definido para F2, etc.

Continuación Lisado 1

```
510 FORI=1TO7STEP2:GOSUB1000:NEXTI
520 FORI=2TO8STEP2:GOSUB1000:NEXTI
530 FORI=9TO12:GOSUB1000:NEXTI
596 REM
597 REM**
598 REM*** CARGAR LA RUTINA DE CODIG
O MAQUINA
599 REM**
600 FORI=SMTOSM+131
610 READB
611 SU=SU+B
612 L=L+1
613 IFL<>19ANDI<>(SM+131)THEN620
614 READCH
615 IFCH=SUTHENL=0:SU=0:GOTO620
616 PRINT:PRINT"[RVSON][SPC]ERROR[SPC]E
N[SPC]SENTENCIAS[SPC]DATA"
617 PRINT"[RVSON]EN[SPC]LA[SPC]LINEA
";PEEK(63)+256*PEEK(64)
618 GOTO800
620 POKEI,B
[illegible]
696 REM
697 REM**
698 REM***PREPARACION DEL PROGRAMA D
E LENGUAJE MAQUINA
699 REM**
700 POKE250,SM/256
710 POKE249,SM-256*PEEK(250)
715 POKE248,76
720 POKE252,SS/256
730 POKE251,SS-256*PEEK(252)
740 POKE649,9
750 POKE55,PEEK(251):POKE56,PEEK(252
)
760 POKE51,PEEK(251):POKE52,PEEK(252
)
761 POKET-13,PEEK(789)
762 POKET-14,PEEK(788)
770 SYS248
780 NEW
796 REM
797 REM**
798 REM***RESTAURAR EL ACTUAL VALOR
DE FIN DE MEMORIA EN CASO DE ERROR
799 REM**
800 POKE56,T/256
810 POKE55,T-256*PEEK(56)
820 END
996 REM
997 REM**
998 REM***SUBRUTINA PARA ALMACENAR L
OS 'STRINGS' DE F$(I)
999 REM**
1000 POKEPP,SP-SS:PP=PP+1
1005 IFF$(I)=""THEN1060
1010 FORJ=1TOLEN(F$(I))
1030 POKESP,ASC(MID$(F$(I),J,1))
1040 SP=SP+1
1050 NEXTJ
1060 POKESP,0
1070 SP=SP+1
1080 RETURN
8996 REM
8997 REM**
8998 REM***DATAS DEL PROGRAMA DE LEN
GUAJE MAQUINA
8999 REM**
9000 DATA120,165,250,141,21,3,165,24
9,24,105,21,144,3,238,21,3,141,20,3,
1837,88
9010 DATA96,165,198,208,92,165,254,2
40,91,169,0,133,254,164,253,24,144,5
6,2794
9020 DATA201,6,240,17,202,201,5,240,
12,202,201,4,240,7,169,255,141,128,2
,2473
9030 DATA48,58,134,253,162,0,173,141
,2,201,1,208,2,162,4,201,2,208,2,196
2,162
9040 DATA8,138,24,101,253,205,128,2,
240,30,141,128,2,168,177,251,168,162
,2488,0
9050 DATA177,251,240,17,157,119,2,20
0,232,134,198,224,9,208,241,132,253,
169,2963
9060 DATA1,133,254,76,49,234,162,3,1
65,197,201,3,240,187,202,24,144,162,
2437
```

Para volver a definir una tecla de función, se modifica la línea de programa que corresponde a la tecla de función que se cambiará.

Por ejemplo, la línea 210 define el "string" para F1:

210F$(1)="LIST"

Si el usuario prefiere imprimir la palabra LOAD cuando se pulsa la tecla F1, la línea 210 se modifica a:

210F$(1)="LOAD"

Es importante incluir las comillas, dado que F$ es una variable de cadena.

Cualquier "string" válido puede ser asignado a las teclas de funciones, incluyendo los "strings" que contienen los controles del cursor. Sin embargo, resulta un poco más complicado asignar los caracteres Return y las comillas dentro de un "string". Para incluir un carácter Return dentro de un "string", se añade CHR$(13) al "string" (13 representa Return en el código ASCII). Por ejemplo, si el usuario desea que un programa se ejecute automáticamente cuando se pulsa la tecla F3, la línea 230 se tiene que convertir en:

230F$(3)="RUN" + CHR$(13)

El retorno será ejecutado justo después de que se imprima RUN, como si la tecla Return hubiese sido pulsada en el teclado. De forma semejante, las comillas se pueden incluir dentro de un "string" mediante el uso de CHR$(34).

La máxima longitud total de todos los "string" asignados a las teclas de funciones es de 231 caracteres. Si se asignan más de 231 caracteres, el programa imprimirá un mensaje de error indicando que los "string" son demasiado largos. En este momento es una simple cuestión de editar las líneas apropiadas y volver a ejecutar el programa.

Hay que tener en cuenta que el programa se borra de la memoria después de ejecutarse. Por lo tanto, es importante salvar el programa antes de que se ejecute si se requiere una copia de éste con las recién definidas funciones.

A continuación se presenta una breve descripción de cómo funciona la sección del lenguaje máquina del programa.

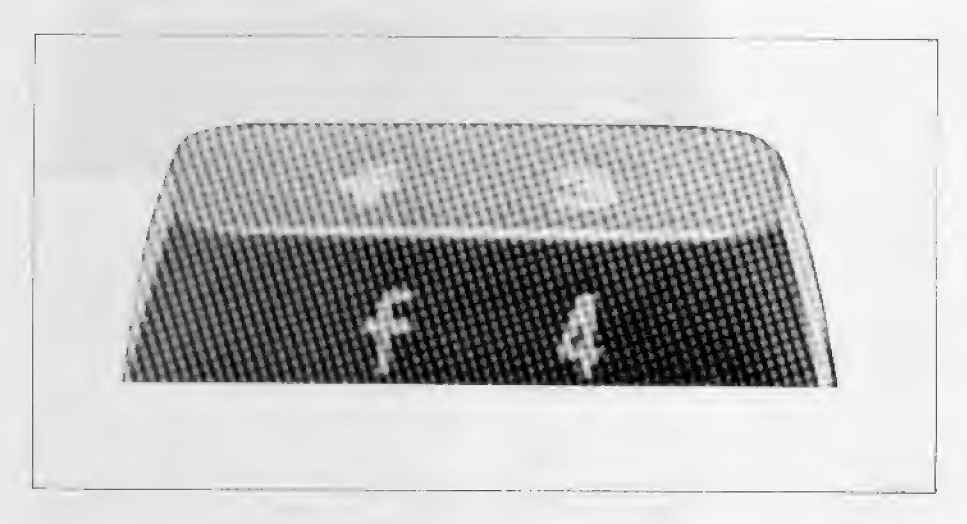
La técnica general que se emplea para activar las teclas de funciones es bastante sencilla. Se genera una interrupción (hardware interrupt) sesenta veces por segundo que obliga al sistema operativo a realizar ciertas funciones internas, como

Listado 2
Teclas de Función VIC-20

```
100 T=PEEK(55)+256*PEEK(56)
110 POKE56,PEEK(46)+2
120 POKE52,PEEK(56)
130 PRINT"[CLR]TECLAS[SPC]DE[SPC]FUN
CION":PRINT
200 DIMF$(12)
210 F$(1)="LIST"
220 F$(2)="LOAD"+CHR$(34)
230 F$(3)="RUN"
240 F$(4)="SAVE"+CHR$(34)+"@:"
250 F$(5)="PRINTPEEK([SPC][4SPC])[6CRSRL]"

260 F$(6)="OPEN15,8,15,"+CHR$(34)
270 F$(7)="PRINTFRE(0)"
280 F$(8)="CLOSE15"
290 F$(9)="GOTO"
300 F$(10)="GOSUB"
310 F$(11)="RETURN"
320 F$(12)="POKE"
330 FORI=1TO12
340 PRINT"F"+RIGHT$(STR$(I),LEN(STR$
(I))-1)+"[SPC]=[SPC]";
345 PRINTCHR$(34)+F$(I)+CHR$(34)
350 NEXTI
400 L=0
410 FORI=1TO12
420 L=L+LEN(F$(I))
430 NEXTI
435 IFL>231THENPRINT"[RVSON]'STRINGS
'[SPC]DEMASIADO[SPC]LARGOS":GOTO800
440 SM=T-120
450 SS=T-L-144
500 PP=SS:SP=SS+12
510 FORI=1TO7STEP2:GOSUB1000:NEXTI
520 FORI=2TO8STEP2:GOSUB1000:NEXTI
530 FORI=9TO12:GOSUB1000:NEXTI
600 FORI=SMTOSM+119
610 READB
611 SU=SU+B
612 C=C+1
613 IFC<>19ANDI<>(SM+119)THEN620
614 READCH
615 IFCH=SUTHENC=0:SU=0:GOTO620
616 PRINT:PRINT"[RVSON][SPC]ERROR[SPC]E
N[SPC]SENTENCIAS[SPC]DATA"
617 PRINT"[RVSON]EN[SPC]LA[SPC]LINEA
";PEEK(63)+256*PEEK(64)
618 GOTO800
620 POKEI,B
630 NEXTI
700 POKE2,SM/256
710 POKE1,SM-256*PEEK(2)
720 POKE252,SS/256
730 POKE251,SS-256*PEEK(252)
740 POKE649,9
750 POKE55,PEEK(251):POKE56,PEEK(252
)
760 POKE51,PEEK(251):POKE52,PEEK(252
)
770 SYS0
780 NEW
800 POKE56,T/256
810 POKE55,T-256*PEEK(56)
820 END
1000 POKEPP,SP-SS:PP=PP+1
1005 IFF$(I)=""THEN1060
1010 FORJ=1TOLEN(F$(I))
1030 POKESP,ASC(MID$(F$(I),J,1))
1040 SP=SP+1
1050 NEXTJ
1060 POKESP,0
1070 SP=SP+1
1080 RETURN
9000 DATA120,165,2,141,21,3,165,1,24
,105,21,144,3,238,21,3,141,20,3,1341

9010 DATA88,96,165,198,208,92,165,25
4,240,9,169,0,133,254,164,253,24,144
,56
9020 DATA2712,162,3,169,63,197,197,2
40,13,56,233,8,202,16,246,169,255,14
1,128
9030 DATA2,2500,48,58,134,253,162,0,
173,141,2,201,1,208,2,162,4,201,2,20
8
9040 DATA2,1962,162,8,138,24,101,253
,205,128,2,240,30,141,128,2,168,177,
251,168
9050 DATA162,2488,0,177,251,240,17,1
57,119,2,200,232,134,198,224,9,208,2
41,132
9060 DATA253,169,2963,1,133,254,76,1
91,234,889
```



el barrido del teclado y la actualización del reloj de tiempo real. Al interceptar esta interrupción, el programa en lenguaje máquina se ejecuta sesenta veces por segundo.

Cada vez que el programa se ejecuta, comprueba para averiguar si una de las teclas de funciones ha sido pulsada. Si una tecla ha sido pulsada, el "keyboard buffer" (memoria intermedia del teclado) se carga con todos los caracteres del "string" apropiado que pueda contener. Una vez que el sistema operativo haya vaciado el "keyboard buffer", mi programa cargará el "buffer" con el resto del "string" hasta que éste se imprima del todo.

El Programa en Basic

Las funciones del programa en Basic son las siguientes: cargar el programa en lenguaje máquina al final de la memoria; cargar los "strings" justo debajo del código máquina; inicializar los punteros para el programa en lenguaje máquina; y proteger el programa y los "strings" del resto del sistema operativo.

Líneas 100 130. Estas líneas reservan la memoria suficiente para cargar el programa en lenguaje máquina y los "strings" cambiando el puntero de final de memoria para que apunte a 512 bytes por encima del programa en Basic.

Líneas 200 a 350. Crean el "array" F$, y se imprimen los "strings" relacionados con cada tecla de funciones para recordarle al usuario cómo han sido definidas.

Líneas 400 a 450. Estas líneas calculan la longitud total de todos los "strings". Se comprueba que su longitud no sobrepase de los 231 caracteres. Si los "strings" son válidos, se calculan el SM (comienzo del programa en lenguaje máquina) y el SS (comienzo de los "strings").

Líneas 500 a 530. Estas líneas cargan los "strings" y una tabla de punteros para los "strings", comenzando en la posición SS.

Líneas 600 a 630. El programa en lenguaje máquina se lee a partir de las sentencias, Data y se carga, comenzando en la posición SM.

Líneas 700 a 780. Se almacena en la memoria un puntero para la rutina de inicialización del lenguaje máquina. El puntero del final de memoria se modifica para que apunte al principio de los "strings", para que sólo le quite al usuario la cantidad de memoria necesaria. El programa salta a la rutina de inicialización del lenguaje máquina, y a continuación, ejecuta una sentencia New.

Líneas 800 820. La ejecución llega a este punto sólo si un error ha tenido lugar, como por ejemplo, la definición de "strings" demasiado largos. Se restaura el final de memoria a su valor original para que toda la memoria se devuelva al sistema operativo antes de parar.





1000 a 1080. Esta es la subrutina que carga un "string" F$(I) en la memoria. También se carga un puntero para el "string" en una tabla.

Conclusión

Este programa resulta realmente útil dado que cada usuario lo puede personalizar para satisfacer sus propias necesidades. Si el usuario dispone de una impresora, por ejemplo, una de las teclas puede ser definida para que produzca una salida impresa mediante la pulsación de una sola tecla. Algunos de los comandos definidos por mí resultan útiles solamente si el usuario dispone de una unidad de discos. El usuario encontrará que este programa resulta muy útil si lo usa para definir las funciones que emplea con más frecuencia.

QUIERE TRABAJAR EN SERIO?
NO RETRASE SU PROGRESO!
JUEGUE CON VENTAJA!



LE DESCUBRIMOS NUESTRO JUEGO
PARA SU COMMODORE 64

CONTABILIDAD, GESTION STOCK, BASES DE DATOS, PRACTICALC, MAILING

PROCESO DE TEXTOS, GENERADOR DE GRAFICOS Y SONIDOS, VIC-TREE,

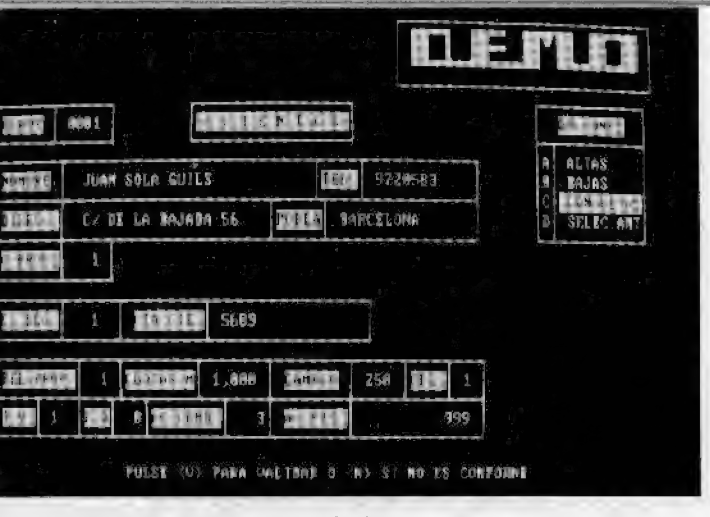
LAPIZ OPTICO, UTILITARIOS, JUEGOS Y ... MUCHOS MAS!!!

pidanos información, recibirá un PROGRAMA DE REGALO y nuestro extenso catálogo.

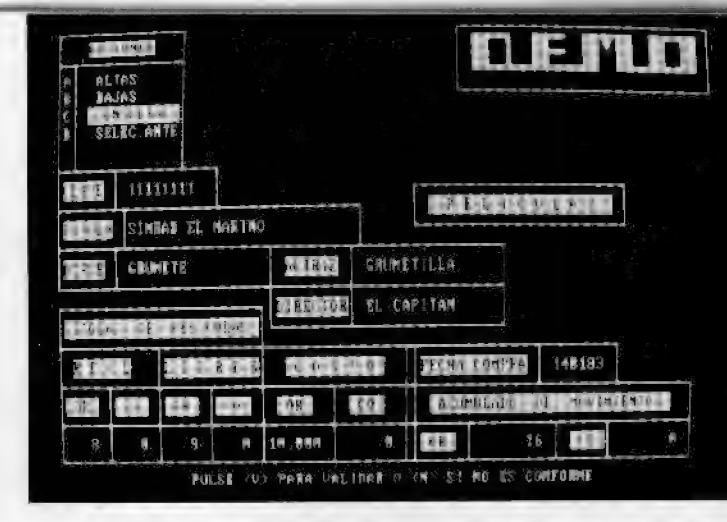
EAF microgestion

consejo de ciento, 563-565
telefono 93-231 95 87

barcelona-13
apdo. 24.143



Base de datos



Base de datos

Fotos cedidas por DEMO

VENTANAS CBM (IV)

READY

(del chip a la base de datos)

(cont.)

Por Rafael NAVARRO



CBM-700

Ficheros de Claves

Los ficheros de claves convierten a los relativos en ficheros indexados pues los primeros son como un directorio o índice de los segundos. Cualquier dato o mezcla de datos de un fichero relativo puede ser controlado por un fichero de claves permitiendo así obtener a través de uno, el resto de los datos de un registro. Así, por poner un ejemplo, conociendo un nombre se obtiene un teléfono y conociendo un teléfono se obtiene una dirección o cualquier combinación de datos. Algunos sistemas permiten utilizar como índices o directorios de un fichero, datos no contenidos específicamente dentro de los registros del fichero. Esta técnica es sumamente peligrosa ya que, en caso de destrucción del fichero de claves o directorio, se hace imposible la recuperación del mismo. Por este motivo, MEC/DOS, solamente permite utilizar directorios basados en datos internos al registro. En caso —improbable— de que se pierda el directorio, un simple comando permite su regeneración automática ya que bastará con repasar los registros del fichero relativo para encontrar (los ficheros de claves disponen de un descriptor que indica dónde se encuentran los datos clave para el directorio) los datos clave y reconstruir el fichero de índices. El mismo comando servirá también para crear nuevos directorios, no previstos en el análisis inicial, a medida que sean necesarios.

El tratamiento de claves incluye las altas, bajas, búsquedas y barridos de ficheros de claves. Un fichero relativo puede tener tantos ficheros de claves como se desee.

Las claves de los ficheros de claves no se encuentran en la ram, si no que están en disco y es allí donde se ordenan y registran.

La capacidad máxima de los ficheros de claves y relativos es de 65535 registros.

Ficheros Tipo Tabla

La gran capacidad y el alto índice de seguridad de los ficheros de claves hacen que, para grandes ficheros, y con

unidades de diskette (no con whinchester), la velocidad de inserción caiga un poco. Por este motivo, MEC/DOS dispone de otro sistema de archivo que mezcla las características estructurales de los ficheros relativos y los de claves con una velocidad de trabajo del orden de centésimas de segundo para cualquiera de las operaciones (borrado, búsqueda o inserción).

Los ficheros tablas residen en disco y son cargados enteros en RAM cuando son requeridos por el programador. La estructura de sus registros puede ser definida con todas las posibilidades que ofrecen los ficheros relativos, pero ofrecen una indiscutible ventaja sobre éstos: intrínsecamente y por definición, TODOS LOS CAMPOS SON CLAVE. Es decir, en cualquier momento, y sin ordenamientos previo —en centésimas de segundo—, se puede acceder a cualquier registro, por cualquier campo. Y no sólo eso: es posible obtener, con los mismos condicionantes de tiempo, una lista ordenada del fichero, utilizando como criterio de ordenación cualquiera de los campos del registro. Esta es una característica única hasta el momento.

Los ficheros de tablas residen fuera del cartucho MEC/DOS, ocupando el área disponible de memoria del sistema 7XX, pero de manera totalmente transparente al usuario. Dada la sencillez de manejo de MEC/DOS, un programa de 5K puede realizar un proceso relativamente complejo, de modo que es posible tener en memoria tablas por un volumen aproximado de 55K con un acceso directo a las mismas. Si a esto se añade el hecho de que MEC/DOS permite empaquetar (reducir la ocupación física) tanto campos numéricos como campos alfa, se comprende la potencia intrínseca del tratamiento de tablas.

Pantallas

MEC/DOS permite, mediante un sencillo utilitario, definir pantallas de trabajo. Dichas pantallas pueden contener gráficos y caracteres alfabéticos simultáneamente. La idea es facilitar la buena presentación de los paquetes de programas. Por otro lado, el generador de pantallas ofrece la posibilidad de definir partes de pantalla como zonas de trabajo sobre las cuales se presentarán datos o se pedirán datos al operador. Asimismo, es posible definir zonas o módulos de pantalla posteriormente miscibles.

Cada una de las zonas de trabajo puede ser definida como alfa, alfa solamente ASCCI o numérica y, en el caso de las numéricas se permite especificar otros parámetros tales como número de decimales, decimales y enteros forzados a cero, signo, etc. Posteriormente, la pantalla o

FICHEROS TABLAS

COMANDO	EXPLICACION	VALORES DE OK
TFIELD, NF, NC	Asigna el campo NC como clave del fichero NF para las búsquedas que posteriormente se efectúen en el fichero.	OK= 0; correcto OK= 1; comando no permitido OK= 5; fichero no asignado OK=14; NC fuera de rango
TPOS, NF, VE$	Busca en la tabla NF una clave igual a VE$. Si lo encuentra, se posiciona, permitiendo operar con el registro. Si no lo encuentra se posiciona sobre el inmediatamente posterior.	OK= 0; operación correcta (clave encontrada) OK= 1; comando no permitido OK= 5; NF no asignado OK=11; registro inexistente
TNEXT, NF	Se posiciona sobre el siguiente registro, dentro de la tabla NF, siguiendo el orden secuencial ascendente según el campo clave.	OK= 0; correcto OK= 1; comando no permitido OK= 5; NF no asignado OK=11; fin de fichero
TREG, NF, NR	Posiciona el puntero de registro de la tabla NF en el registro NR.	OK= 0; correcto OK= 1; comando no permitido OK= 5; NF no asignado OK= 6; NR fuera de rango
TPOINT, NF, NR	Devuelve en NR el número de registro al que apunta NF	OK= 0; correcto OK= 1; comando no permitido OK= 5; NF no asignado
TERASE, NF	Elimina de la tabla NF el registro en curso.	OK= 0; correcto OK= 1; comando no permitido OK= 5; NF no asignado
TFIRST, NF	Posiciona el puntero de registro de la tabla NF sobre el primer registro libre. Este comando, en combinación con el comando BUFPUT, permite dar de alta nuevos registros.	OK= 0; correcto OK= 1; comando no permitido OK= 5; NF no asignado OK= 8; fichero completo

FICHEROS DE CLAVES

COMANDO	EXPLICACION	VALORES DE OK
SEARCH, NF, CL$, NR	Busca la clave CL$ en el fichero de claves NF, devolviendo la dirección del registro del fichero indexado en NR. Si no encuentra CL$, NR vale cero.	OK= 0; correcto OK= 5; NF no asignado OK=11; CL$ no encontrada
SNEXT, NF, CL$, NR	Entrega en CL$ y NR la clave y dirección actualmente indicados por el puntero del fichero de claves, incrementado posteriormente el mismo. Nótese que funciona de modo diferente al comando de tablas TNEXT.	OK= 0; correcto OK= 5; NF no asignado OK=11; fin de fichero NOTA: OK=11 tiene significados diferentes según el comando que lo provoca
INSERT, NF, CL$, NR	Inserta la clave CL$ en el fichero de claves NF, asignándole NR como dirección del fichero indexado.	OK= 0; correcto OK= 5; NF no asignado OK=10; overflow de claves (fichero lleno)
DELETE, NF, CL$, NR	Borra del fichero de claves NF, la clave CL$, cuya dirección asignada es NR (se permiten claves duplicadas)	OK= 0; correcto OK= 5; NF no asignado OK=11; CL$/NR inexistente
KEYDATA, NF, PB, NB	Devuelve en PB y NB el primer byte y el número de bytes del registro del fichero indexado que constituyen la cadena clave para indexar en el fichero NF.	OK= 0; correcto OK= 1; comando no apropiado OK= 5; NF no asignado
REORG, NF	Reorganiza, distribuyendo uniformemente sus claves, el fichero de claves NF.	OK= 0; correcto OK= 5; NF no asignado
GENCLA, NFC, NFR	Genera sobre NFC un fichero de claves que indexan los registros de NFR según las características de clave que se expresaron en el utilitario generador de ficheros.	OK= 0; correcto OK= 5; NF no asignado OK=10; overflow de claves (fichero lleno)

TRATAMIENTO DE CAMPOS

COMANDO	EXPLICACION	VALORES DE OK
BUFGET, NF, NC, VR$/VR [VR$/VR, VR$/VR...]	Recibe en VR$ o VR, a partir del campo NC, los valores de los campos NC, NC+1, NC+2, etc. recogiéndolos del buffer del fichero NF.	OK= 0; correcto OK= 5; NF no asignado OK=14; NC fuera de rango (o incrementos de NC) OK=18; contenido de campo en mal estado
BUFPUT, NF, NC, VE$/VE [VE$/VE, VE$/VE...]	Escribe en el buffer del fichero NF, a partir del campo NC, los valores de los campos NC, NC+1, NC+2, según los contenidos de VR$/VR, realizando automáticamente, lo mismo que BUFGET, los tratamientos necesarios (compactado en este caso, descompactado en el otro).	OK= 0; correcto OK= 5; NF no asignado OK=14; NC fuera de rango (o incrementos de NC) OK=18; contenido de campo en mal estado

las partes de pantalla podrán ser llamadas desde cualquier programa y se tendrá acceso a las siguientes funciones para cada zona:

—edición de un mensaje
—reverse del contenido de una zona
—borrado del contenido de una zona
—entrada, por parte del operario, de un dato en una zona
—recogida en una variable del dato contenido por una zona.

Para los casos en que no se desee trabajar con pantallas definidas, es posible emplear comandos especiales de acción similar que, en vez de referirse a una zona, indican al sistema la fila y la columna de trabajo, así como otros datos imprescindibles.

A través del interpretador, es posible producir en cualquier momento el volcado sobre impresora del contenido de la pantalla. Si se utilizan impresoras COMMODORE, se reflejarán incluso los gráficos y los caracteres en negativo. Si se desea ofrecer esta opción al operador, sin necesidad de ejecutar un comando específico, también es posible programar —mediante un simple comando— el sistema de interrupciones, para activar la función de volcado en cualquier momento a la pulsación de la tecla "F10".

Múltiple Precisión

El paquete matemático permite operar en coma flotante y con tratamiento de signo, valores de hasta 22 dígitos numéricos. Dado que el interpretador BASIC de COMMODORE pasa a notación científica valores de más de 9 cifras, los operadores y el resultado deberán ser soportador por variables de cadena. Se ofrece asimismo un comando de formatación de resultados para el tratamiento del signo y comas de millar. Las operaciones soportadas son la suma, la resta, la multiplicación y la división.



TRATAMIENTO DE CAMPOS

COMANDO	EXPLICACION	VALORES DE OK
BUFMID, NF, PB, NB, VR$	Recoge, del buffer del fichero NF, a partir de PB, NB bytes y los asigna a VR$. Usése en combinación con KEYDATA.	OK= 0; correcto OK= 1; comando no apropiado OK= 5; NF no asignado

PANTALLAS

COMANDO	EXPLICACION	VALORES DE OK
ENTER, LI, C, LD, TP, VE$, VR$	Adquisición de VR$ en la línea LI, apareciendo previamente el mensaje VE$ en la columna C. La longitud máxima admitida es de LD caracteres y el tipo de dato, TP (0=alfa, 1=solo caracteres sin SHIFT, 2=numer.)	OK= 0; correcto OK=19; parámetros incorrectos
DISPLAY, LI, C, VE$	Imprime el mensaje VE$ en la línea LI, columna C.	OK= 0; correcto OK=19; parámetros incorrectos
ACTIV, [T(VE)] [,B(VE)] [,L(VE)] [,R(VE)]	Definición de pantalla activa. Delimita por arriba (T), abajo (B), izquierda (L) y derecha (R). Equivalente a los pokes en 220, 221, 222 y 223. Ver manual de BASIC de COMMODORE	OK= 0; correcto OK=19; parámetros incorrectos
DUMPSI [,NP]	Habilitación de la posibilidad de obtener un volcado de la pantalla en la impresora, pulsando "F10". Opcionalmente puede expresarse el número de periférico de la impresora, que por defecto será 4.	OK= 0; correcto
DUMPNO	Antagonista de DUMPSI	OK= 0; correcto
DUMP	Vuelca pantalla sin alterar interrupciones	OK= 0

ZONAS DE PANTALLAS

COMANDO	EXPLICACION	VALORES DE OK
CHARGE, NF$ [,U(PE)] [D(DR)] [,P]	Carga el descriptor de la pantalla NF$, del drive DR unidad PE y, si se escribe con P, carga, además, la pantalla. Los valores por defecto son cero para DR y 8 (o lo indicado en DISK) para PE.	OK= 0; correcto OK=130; memoría agotada
ACCESS, NZ, TV [,NZ,TV,NZ,TV...]	Entrada de un dato en la zona NZ con soporte editor, filtro de los caracteres entrados según el tipo de zona y formatación posterior de los mismos. No entrega el dato en ninguna variable, sólo permite su introducción en la zona. Permanecen activas las siguientes Teclas de Control: DEL: Borrado del carácter donde se halla el cursor. INST: Apertura del texto para insertar un carácter. =CRSR= : Desplaza el cursor a la derecha, o a la izquierda con SHIFT. HOME: Envía el cursor a inicio de zona. CLR: Borra contenido de la zona. TAB: Avanza cinco columnas, o las retrocede con SHIFT. C= (logo Commodore): Se posiciona a la derecha del último carácter escrito de la zona. SHIFT/C=: Borra desde la posición del cursor hasta el final.	OK= 0; correcto OK=16; NZ inexistente
ZOUT, NZ, VE$ [,NZ,VE$,NZ, VE$...]	Imprime VE$ formateada según definición en el módulo de creación de pantallas, sobre la zona NZ.	OK= 0; correcto OK=16; NZ inexistente
ZGET, NZ, VR$ [,NZ,VR$,NZ, VR$...]	Recibe en VR$ el contenido de la zona NZ.	OK= 0; correcto OK=16; NZ inexistente
ZREV, NZ [,NZ,NZ...]	Conmuta REVERSE/NORMAL NORMAL/-REVERSE el contenido de NZ.	OK= 0; correcto OK=16; NZ inexistente
ZCLEAR, NZ [,NZ, NZ]	Borra el contenido de la zona NZ.	OK= 0; correcto OK=16; NZ inexistente
INKEY, VR$ [,VE$] [,NL,NL,...]	Espera la pulsación de una tecla, opcionalmente filtrada por VE$, la coloca en VR$ y salta opcionalmente a la línea NL, según el valor ASCII de VR$ respecto VE$	OK= 0

MULTIPLE PRECISION

COMANDO	EXPLICACION	VALORES DE OK
ADD, VE$, VE$, VR$	Suma en VR$, VE$+VE$. VE$ debe ser una cadena numérica: + 12311111234.234, por ejemplo. Las rutinas de cálculo consideran el signo de sus operandos. Las cadenas nulas valen cero.	OK= 0; correcto OK=18; datos incorrectos

MULTIPLE PRECISION

COMANDO	EXPLICACION	VALORES DE OK
SUB, VE$, VE$, VR$	Coloca en VR$ el resultado de restar del primer operando el segundo.	OK= 0; correcto OK=18; datos incorrectos
MULT, VE$, VE$, VR$	Coloca en VR$ el producto de las variables de entrada.	OK= 0; correcto OK=18; datos erróneos
DIV, VE$, VE$, VR$	Coloca en VR$ el resultado de dividir el primer operando por el segundo.	OK= 0; correcto OK=18; datos erróneos
FORMAT, VE$, VS$, LD, ND, FS	Formata la variable VE$ (que se supone contiene un dato numérico de múltiple precisión o no) sobre la variable VS$, a una longitud de LD caracteres, con ND decimales. Si la variable FS vale 0, se colocará signo sólo si el valor es positivo, si vale 1 se colocará en cualquier caso, y si vale 2, en ningún caso.	OK= 0; correcto OK=11; datos incorrectos

GESTION DE ERRORES

COMANDO	EXPLICACION	VALORES DE OK
ERRSI	El interprete MEC/DOS, detiene la ejecución de MEC/DOS, imprimiendo un mensaje de error y listando la línea donde se produce, siempre que el error sea de sintaxis o insuperable. Este comando permite que, además, siempre que se produzcan valores de OK que indiquen error se produzca el proceso descrito. En todo caso, el valor de OK para errores de sintaxis o propios del interprete es 127.	OK= 0; correcto
ERRNO	Antagonista del anterior comando que, no impide, sin embargo, la interrupción del programa ante errores insuperables.	
OK	Si MEC/DOS se halla en modo ERRNO, y en cualquier caso, si se desea conocer el valor de OK y la descripción del posible error, utilice este comando.	OK= mismo valor que tuviera antes de emplear el comando

En este juego para el C-64, tienes que matar a unos mosquitos rápidos. Conseguirás una buena puntuación si "mosqueas" a muchos.

"Mosqueando al Mosquito" es un juego simple pero divertido, escrito totalmente en Basic para el C-64. La finalidad del juego es la de matar al mosquito. Esto se hace moviendo la palmeta mediante el uso del "joystick" conectado a la Puerta 1. El mosquito se desplaza al azar (igual que en la vida real) por toda la pantalla. Cuando intentas darle al mosquito, sólo tienes una posibilidad entre nueve de matarlo (igual que en la vida real). Cuando matas a dos mosquitos, los otros se "mosquean", y los mosquitos grandes empiezan a aparecer al azar por toda la pantalla. Si tú o el mosquito pequeño chocáis con un mosquito grande, los dos seréis desplazados a distintas posiciones en la pantalla. A veces, tú puedes sacar ventaja de esto, si tu palmeta se encontraba lejos del mosquito pequeño.

¡Pero ojo! Uno de los mosquitos grandes es un matón. Si tú o el mosquito pequeño le tocáis a éste, el juego habrá terminado. No hay forma de saber cuál es el mosquito matón hasta que choques con él. Si le das, el mosquito matón se vuelve rojo, y tu puntuación saldrá en pantalla.

C-64
JOYSTICK

Cuando hayas matado a diez mosquitos pequeños, el juego termina, y el tiempo que hayas tardado sale en pantalla. Cualquier puntuación por debajo de los 100 segundos, es buena.

Sonido

El programa comienza con el título, y un mensaje de inicialización. Durante estos mensajes previos, el ordenador se encuentra muy ocupado transfiriendo el juego de caracteres a la memoria RAM. Esto le permite al programa crear los caracteres del juego.

Las líneas 130-160 crean el carácter para el mosquito pequeño y el carácter para la palmeta. Las líneas 1000-1060 crean los "sprites" para los mosquitos grandes. Existen dos efectos de sonido en el programa: el "zzzz" del mosquito y el sonido del palmetazo. Estos sonidos están creados en las líneas 260-330 y 900-980, respectivamente. Las líneas 560-580 crean los movimientos aleatorios del mosquito.

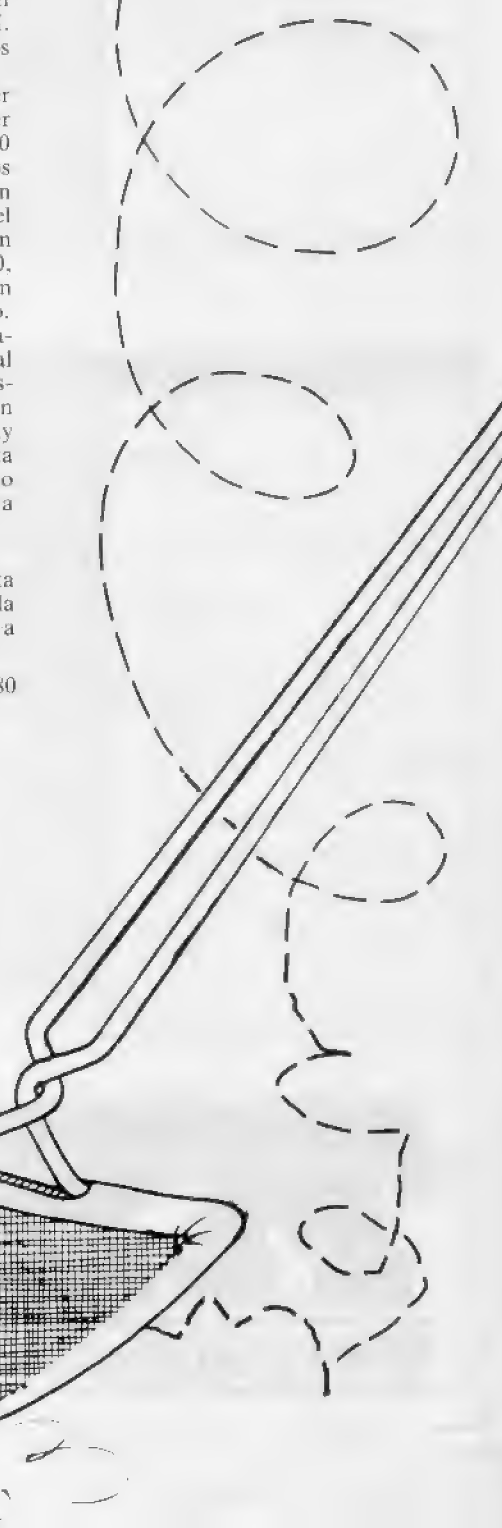
Cuando el ordenador termina la inicialización, el jugador comienza el juego al pulsar el botón de disparo en el "joystick". **No** es necesario pulsar este botón para matar al mosquito. Lo único que hay que hacer es intentar mantener la palmeta encima del mosquito, y tarde o temprano será matado. Pero es importante seguir a los pequeños y evitar a los grandes.

¡Animo! Los matarás bien muertos.

(El juego se hace más rápido si se quita el sonido del mosquito, cambiando la línea 630 por la que se representa a continuación:

```
630 POKEMD,32:POKEMC,0:GOTO380
```

Mosqueando al Mosquito





```
10 PRINT"[CLR]":FORI=1TO5:PRINT:NEXT
:PRINTTAB(16);"MOSQUITO"
15 PRINT:PRINT:PRINTTAB(19);"POR"
20 PRINT:PRINT:PRINTTAB(12);"CHARLES
[SPC]T.[SPC]KOWAL":N=0:Q=160:R=110:G
=1
30 POKE53269,0:POKE53277,1:POKE53271
,1:POKE53279,0:GOSUB1000
35 REM*** TRANSFIERE EL JUEGO DE CAR
ACTERES A RAM ***
40 POKE56334,PEEK(56334)AND254:POKE1
,PEEK(1)AND251
50 FORI=0TO263:POKEI+12288,PEEK(I+53
248):NEXT
60 PRINT"[CLR]PREFIJANDO[SPC]LA[SPC]T
ABLA[SPC]DE[SPC]CARACTERES"
70 FORI=384TO495:POKEI+12288,PEEK(I+
53248):NEXT
80 FORI=816TO823:POKEI+12288,PEEK(I+
53248):NEXT
90 FORI=1024TO1287:POKEI+12288,PEEK(
I+53248):NEXT
100 FORI=1408TO1519:POKEI+12288,PEEK
(I+53248):NEXT
110 POKE1,PEEK(1)OR4:POKE56334,PEEK(
56334)OR1
120 POKE53272,(PEEK(53272)AND240)+12

125 REM*** FORMAR EL CARACTER DEL MO
SQUITO ***
130 M=12288:POKEM,144:POKEM+1,80:POK
EM+2,50:POKEM+3,252:POKEM+4,82:POKEM
+5,73
140 POKEM+6,136:POKEM+7,132
145 REM*** HACER 'PLAFT'  ***
150 M=M+1248:POKEM,137:POKEM+1,74:PO
KEM+2,36:POKEM+3,0:POKEM+4,0:POKEM+5
,36
160 POKEM+6,74:POKEM+7,137
170 PRINT"[CLR]"
180 PRINT"PULSE[SPC]DISPARO[SPC]PARA
[SPC]EMPEZAR"
190 IFPEEK(56321)<>239THEN190
200 POKE53269,0:POKE53277,PEEK(53277
)AND254 POKE53271,PEEK(53271)AND254
210 POKE53279,0:YY=INT(8*RND(0)):ZZ=
2↑YY
220 PRINT"[CLR]":FORN=56256TO56295:P
OKEN,14:NEXT
225 FORN=55296TO56255:POKEN,1:NEXT
230 FORN=1948TO2023:POKEN,160:NEXT:R
EM*** BORRAR LA LINEA INFERIOR ***
240 SC=1524:DX=0:DY=0 SD=1024:CT=0:G
=0 TM=TI/60
250 MD=MC:POKEMD,32:MC=1024+INT(960*
RND(0)):POKEMC,0:POKESC,102
255 REM***  SONIDO DEL MOSQUITO  ***

260 S=54272:FL=0
270 POKES+24,0
280 POKES+1,100
290 POKES+5,219
300 POKES+15,28
310 POKES+24,15
320 POKES+4,19
330 FORT=1TO200:NEXT:POKES+4,18
375 REM***  LEER JOYSTICK   ***
380 JV=15-(PEEK(56321)AND15):DX=0:DY
=0
400 IFJV=1THENDY=-40:POKESD,32:GOTO4
90
410 IFJV=2THENDY=40:POKESD,32:GOTO49
0
420 IFJV=4THENDX=-1:POKESD,32:GOTO49
0
430 IFJV=5THENDX=-1:DY=-40:POKESD,32
:GOTO490
440 IFJV=6THENDX=-1:DY=40:POKESD,32:
GOTO490
450 IFJV=8THENDX=1:POKESD,32:GOTO490

460 IFJV=9THENDX=1:DY=-40:POKESD,32:
GOTO490
470 IFJV=10THENDX=1:DY=40:POKESD,32
490 SC=SC+DX+DY:IFSC>1983THENSC=SC-9
60
500 IFSC<1024THENSC=SC+960
510 POKESC,102:SD=SC
520 XX=PEEK(53279):IFXX=ZZTHENPOKE53
287+YY,2:XX=0:POKE53269,ZZ:GOTO800
530 IFXX<>0THENPOKESD,32:SC=SC+INT(1
0*RND(0)):SD=SC:POKESC,102:XX=0:GOTO
250
540 POKES+4,19:FORT=1TO10:NEXT:POKES
+4,18
550 DX=0:DY=0
560 MX=INT(RND(0)*3):MX=MX-1
570 MY=INT(RND(0)*3):MY=(MY-1)*40
580 MD=MC:MC=MC+MX+MY
590 IFMC>1983THENMC=MC-960
600 IFMC<1024THENMC=MC+960
610 IFMC=SCTHENTN=TI/60:GOTO900
620 POKES+4,19:FORT=1TO10:NEXT:POKES
+4,18
630 POKEMD,32:POKEMC,0:GOTO280
700 CT=CT+1
710 IFCT>9THENPOKE53269,0:GOTO800
720 IFCT<2THEN740
730 N=CT-2:RESTORE:Q=155*RND(0):R=17
5*RND(0):G=2*G+1:GOSUB1040
740 POKE1984,147:POKE1985,131:POKE19
86,143:POKE1987,146:POKE1988,133
750 POKE1989,189:POKE1990,(CT+176)
760 POKESC+54272,1:GOTO250
800 POKEMD,32:POKEMC,32
810 PRINT"[CLR]","[RVSON]FIN[SPC]DE[SPC]J
UEGO[RVSOFF]":PRINT"TIEMPO=";INT(TN-
TM+.5);"SEGUNDOS"
820 PRINT"PUNTUACION=";CT:POKE53279,
0:POKES+24,0
830 PRINT"[5CRSRD]PULSE[SPC]DISPARO[SPC]P
ARA[2SPC]COMENZAR[SPC]OTRA[SPC]VEZ"
FL=1:GOTO190
895 REM*** SONIDO DE LA PALMETA ***
900 FORL=0TO24:POKES+L,0:NEXT:POKESC
,156:POKEMD,32:POKESC+54272,7
910 POKES,240:POKES+1,33
920 POKES+5,8
930 POKES+22,104
940 POKES+23,1
950 POKES+24,79
960 POKES+4,129
970 FORT=1TO250:NEXT:POKES+4,128
980 FORL=0TO24:POKES+L,0:NEXT
990 GOTO700
1000 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,9
7,0,0,99,0,0,54,0,0,54,0,0,28,0,1,25
5,64
1010 DATA3,255,224,0,218,32,1,85,16,
2,84,136,2,84,128,4,146,128,0,0,0,0,
0,0,0,0
1020 DATA0,0,0,0,0,0,0
1030 FORI=2040TO2047:POKEI,13:NEXT
1040 FORI=0TO62:READZ:POKE832+I,Z:NE
XT
1050 V=53248:POKEV+2*N,R+50:POKEV+1+
2*N,Q+40
1060 POKEV+16,0:POKEV+21,G:POKEV+39+
N,1
1070 IFPEEK(53279)<>0THENMD=MC:POKEM
D,32:MC=MC+5:SD=SC:POKESD,32:SC=SC+5

1080 RETURN
```



CROINFORMATICA
LES CASTELLANA

En la 6ª Planta de El Corte Inglés Castellana, le esperan 1.000 m² dedicados de lleno a la Microinformática. Con las principales marcas del mercado de los Microordenadores Profesionales y Microordenadores Familiares, áreas dedicadas a Complementos y Librería, actividades para colegiales... y todos los Servicios de una gran Feria. Además, un equipo humano, especializado en los diferentes Microordenadores, contestarán a todas sus consultas.

Y no olvide nuestras facilidades de compra: Financiación hasta 24 meses. Con el Servicio Post-Venta y la garantía de El Corte Inglés.
Venga a conocer las novedades de las mejores marcas, en la 1ª Feria de la Microinformática en El Corte Inglés.



6ª P L A N T A - C A S T E L L A N A



Compartiendo Experiencias entre amigos

Esta sección está dedicada a la colaboración de todos nuestros lectores y está dividida en dos partes:

1) Programación:
Programas y similares

2) Magia:
Trucos, sugerencias, etc.

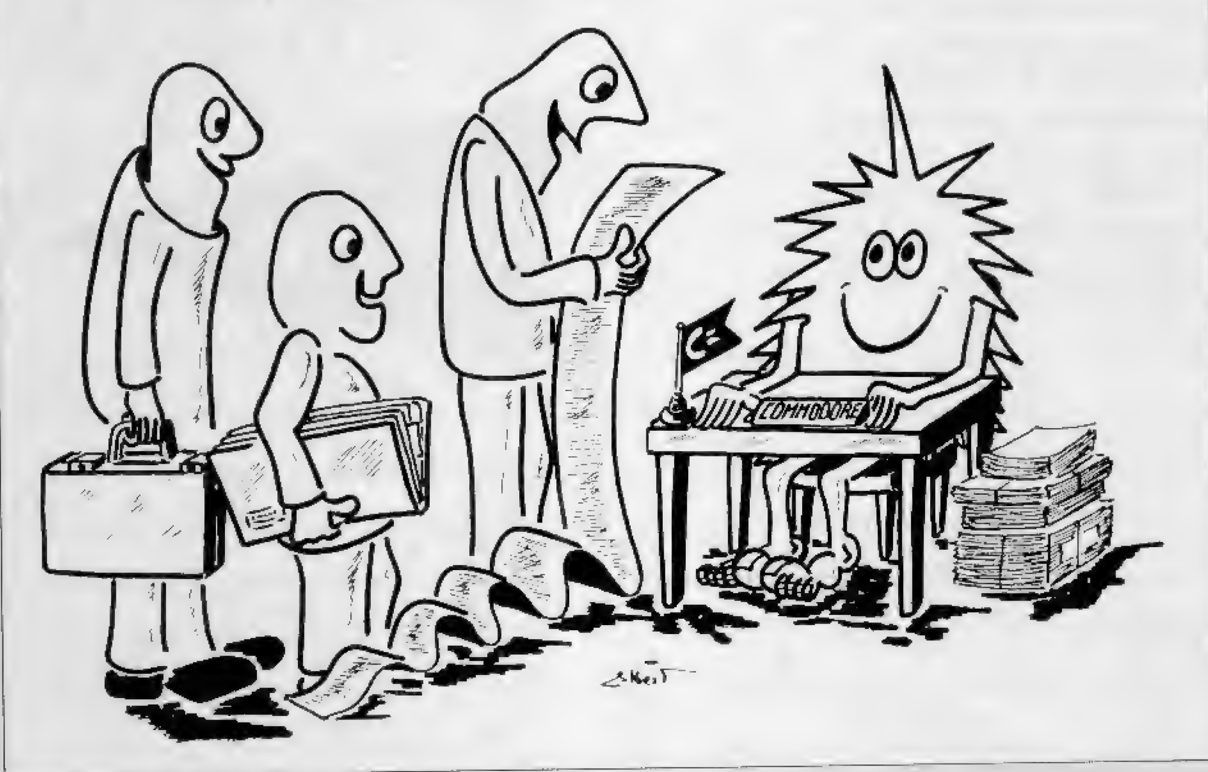
Habrá premios y alicientes "para todos los participantes" (ver editorial página 3).

Todas las colaboraciones deben venir escritas a máquina a doble espacio y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano pero con letra muy clara.

Enviarnos vuestra dirección para que podáis poneros en contacto unos con otros.

Incluimos hoy dos colaboraciones de dos chavales a ver si a los demás se les quita el miedo y nos envían cosas. Topo Loco ya apareció anteriormente en Club Commodore.

Así que: ¡animaros, chicos (...y chicas)!





Golf

Carlos Pérez Orue
Aretxondo, 8-1º D
Algorta-Guecho
VIZCAYA

Hola, amigos. Os mando en la cinta un juego muy corto, pero que espero os guste. Se llama GOLF, y os lo mando en cinta porque no tengo impresora.

El programa está hecho en un Commodore-64, y os lo voy a comentar un poco, aunque es de lo más simple:

100-140: Dibujan el campo de golf.
150-160: Dibujan el hoyo en un punto elegido al azar.
170-180: Dibujan la pelota.
200-240: Movimiento de la pelota.
290-340: Dibujan la pelota que cae al hoyo.

Y nada más, tengo 16 años.

```
1 REM GOLF
10 REM *****************
20 REM *               *
30 REM *     GOLF      *
40 REM *               *
50 REM *****************
90 V=1024:W=55296
100 PRINT"[CLR]":REM "[CLR]"ES SHIFT
 CLR/HOME
110 FOR X=0 TO 39
120 POKE V+(40*20)+X,69
130 POKE W+(40*20)+X,1
140 NEXT
145 PRINT"EL[SPC]CAMPO[SPC]TIENE[SPC]
40[SPC]METROS"
150 CL=INT(39*RND(1)+1)
160 POKE V+(40*20)+CL,32
170 POKE V+(40*19),81
180 POKE W+(40*19),0
190 INPUT "A[SPC]CUANTOS[SPC]METROS[SPC]
ESTA[SPC]EL[SPC]HOYO";A
195 IFA>39 THENA=39
196 IFA<1THENA=0
200 FOR H=1 TO A
205 POKE V+(40*19),32
210 POKE V+(40*19)+H,81
220 POKE W+(40*19)+H,0
230 POKE V+(40*19)+H,32
240 NEXT H
250 POKE V+(40*19)+A,81
260 POKE W+(40*19)+A,0
270 IF A=CL THEN 290
280 GOTO320
290 POKE V+(40*21)+A,81
300 POKE W+(40*21)+A,0
310 POKE V+(40*19)+A,32
315 PRINT:PRINT:PRINT:PRINT"[4SPC]O.
K.,LO[SPC]HAS[SPC]CONSEGUIDO"
317 GOTO 330
320 PRINT:PRINT: PRINT"[2SPC]MALA[SPC]
SUERTE,OTRA[SPC]VEZ[SPC]SERA"
330 PRINT:PRINT: INPUT "QUIERES[SPC] JUGAR
[SPC]DE[SPC]NUEVO[SPC](1[SPC]PA RA[SPC]SI)";B
340 IF B=1 THEN 100
345 IF B<>1 THEN PRINT"[CLR]"
350 END
```

Las Siete y Media

CLUB ALAIZ
Plaza Monasterio Santa Gemma, s/n.
Teléfonos 25 44 80 - 25 77 04
PAMPLONA

Le enviamosun programa que ejecuta el famoso juego de las siete y media. A través de Rem quedan explicadas las distintas partes del programa. El ordenador siempre actúa de banca. La posibilidad de juegadores es de hasta 10. De todos modos basta dimensionar la variable D(t) para que puedan jugar más. El número que pide el ordenador al principio del programa es a modo de barajar las cartas.

El programa ha sido elaborado por miembros del Club de Programación Alaiz, de Pamplona, destinado a estudiantes de BUP y COU. Se organizan todo tipo de cursos, actividades, computer camp's, etc.

Este programa está destinado a usarlo con el Commodore 64. Su utilización en un VIC-20 es fácilmente adaptable.

Jesús Rafael Martínez

```
1 REM ***JUEGO SIETE Y MEDIO PARA C-
64***
2 REM REALIZADO POR EL CLUB DE PROGR
AMACION ALAIZ.
3 REM PLZA STA GEMMA S/N PAMPLONA TF
NOS--(948)254480 Y 257704
4 REM ++++AUTORES:DANI,EMILIO,SANTI,
CAMILO,PABLO,KESUS Y JESUS+++++++
16 PRINT"[CLR]"
17 PRINT:PRINT:PRINT:PRINT:PRINT
18 PRINT"[4SPC]7.57.57.5[5SPC]7.57.5
7.5[SPC]"
19 PRINT"[9SPC]7.5[6SPC]7.5[7SPC]"
20 PRINT"[8SPC]7.5[7SPC]7.5[7SPC]"
21 PRINT"[7SPC]7.5[8SPC]7.57.57.5[SPC]
"
22 PRINT"[7SPC]7.5[14SPC]7.5[SPC]"
23 PRINT"[7SPC]7.5[14SPC]7.5[SPC]"
24 PRINT"[7SPC]7.5[3SPC]7.5[2SPC]7.5
7.57.5[SPC]":FORI=1TO2000:NEXTI
25 PRINT"[7SPC]7.5[3SPC]7.5[2SPC]7.5
7.57.5[SPC]":FORI=1TO2000:NEXTI
26 PRINT"[CLR]":PRINT"ESTE[SPC]ES[SPC]
EL[SPC]JUEGO[SPC]DEL[SPC]SIETE[SPC]Y
[SPC]MEDIO,[5SPC]PRUEBEN[SPC]SU[SPC]
HABILIDAD"
27 PRINT"EL[SPC]ORDENADOR[SPC]REPART
E,[SPC]Y[SPC]POR[SPC]TANTO[SPC]ES[SPC]
LA[SPC]BANCA"
28 PRINT"SI[SPC]USTED[SPC]TIENE[SPC]
LA[SPC]MISMA[SPC]JUGADA[SPC]QUE[SPC]
EL,[2SPC]PIERDE[SPC]USTED"
29 FORT=1TON D(T)=0:NEXT:INPUT"ESCRI
BE[SPC]UN[SPC]NUMERO";I
30 INPUT"NUMERO[SPC]DE[SPC]JUGADORES
:[SPC]",N GOSUB700
31 FORT=1TON
```



```
32 PRINT"[CLR]"
33 PRINT"[YEL]JUEGA[SPC]";A$(T)
34 PRINT"[WHT]"
35 REM  * SACA CARTA *
36 GOSUB200
37 REM * JUGADA DE LOS JUGADORES *
38 A=X
39 D(T)=D(T)+A
40 PRINTTAB(3)"[GRN]TIENES[SPC]";D(T
)
41 PRINT"[WHT]"
50 IFD(T)>7.5THENPRINT"SE[SPC]HA[SPC]
PASADO"
62 IFD(T)>7.5THENGOTO91
70 PRINT"QUIERES[SPC]MAS[SPC]CARTAS,
[SPC]";A$(T);"?"
71 PRINT"[2SPC]SI[SPC]QUIERE[SPC]MAS
[SPC]PULSE[SPC]LA[SPC]TECLA[SPC]S,[SPC]
SI[SPC]NO[SPC]PULSE[SPC]OTRA[SPC]TEC
LA[SPC]Y[SPC]ESPERE"
72 GETA$:IFA$=""THEN72
80 IFA$="S"THENGOTO36
91 FORI=1TO2000:NEXTI
92 NEXTT
93 REM * JUEGA EL ORDENADOR *
94 PRINT"[CLR]"
95 G=0
96 PRINT"[BLK]JUEGO[SPC]YO":PRINT"[WHT]
"
100 GOSUB200
105 B=X
110 G=G+B
120 PRINTTAB(3)"TENGO[SPC]";G
130 IFG>5ANDG<7.5THENGOTO150
135 IFG>7.5THENPRINT"ME[SPC]HE[SPC]P
ASADO"
136 IFG>7.5THENGOTO150
140 IFG<=5THENGOTO100
150 FORI=1TO2500:NEXTI
151 REM * COMPARACION *
155 PRINT"[CLR]"
170 FORT=1TON
175 IFD(T)>7.5THEND(T)=0
176 IFG>7.5THENG=0
179 PRINT:PRINT:PRINT:PRINT:PRINT:PR
INT
180 IFD(T)>GTHENPRINTCHR$(18)TAB(5)"
HA[SPC]GANADO[SPC]";A$(T)
181 IFD(T)>GTHENGOSUB490
185 FORI=1TO2050:NEXTI
189 PRINT:PRINT:PRINT:PRINT:PRINT:PR
INT
190 IFG>=D(T)THENPRINTCHR$(18)TAB(5)
"APRENDA[SPC]A[SPC]JUGAR,[SPC]";A$(T
)
191 IFG>=D(T)THENGOSUB300
192 FORY=1TO500:NEXTY:PRINT"[CLR]"
193 NEXTT
194 PRINT"[CLR]":PRINT"QUIEREN[SPC]V
OLVER[SPC]A[SPC]JUGAR?"
195 PRINT"[3SPC]SI[SPC]QUIEREN[SPC]P
ULSEN[SPC]LA[SPC]TECLA[SPC]S,SI[SPC]
NO[3SPC]PULSEN[SPC]OTRA[SPC]TECLA[SPC]
"
196 GETB$:IFB$=""THEN196
197 IFB$="S"THENGOTO29
198 END
199 REM *BARAJA*
200 X=INT(10*RND(I))+1
201 REM * MUSICA SI GANA EL ORDENADO
R *
202 IFN>5THENL=8
203 IFN<=5THENL=4
205 T(X)=T(X)+1:IFT(X)>LTHENGOTO200
210 ONXGOTO220,220,220,220,220,220,2
20,230,240,250
220 PRINT"HA[SPC]SALIDO[SPC]"X:GOTO2
60
230 PRINT"HA[SPC]SALIDO[SPC]J":GOTO2
60
240 PRINT"HA[SPC]SALIDO[SPC]Q":GOTO2
60
250 PRINT"HA[SPC]SALIDO[SPC]K"
260 IFX>7THENX=.5
270 RETURN
300 POKE54296,15
310 POKE54277,3
320 POKE54276,33
330 FORD=1TO100:NEXTD
340 READA
350 READB
360 IFB=-1THENPOKE54273,0:POKE54272,
0:RESTORE:RETURN
370 POKE54273,A:POKE54272,B
380 POKE54276,0
390 GOTO310
400 DATA17,37,19,63,21,154,22,227,17
,37,63,154,22,227
410 DATA52,177,28,214,32,94,34,175
420 DATA-1,-1
430 RETURN
435 REM * MUSICA SI PIERDE EL ORDENA
DOR *
490 FORE=1TO7
500 POKE54296,15
510 POKE54277,55
520 POKE54276,37
530 FORH=1TO300:NEXT
550 POKE54273,17
560 POKE54276,0
590 NEXTE
620 RETURN
625 REM * INTRODUCCION DE NOMBRES *
700 PRINT"[CLR]"
701 IFN=1THENPRINT"ESCRIBA[SPC]SU[SPC]
NOMBRE"
702 IFN>1THENPRINT"ESCRIBAN[SPC]SUS[SPC]
NOMBRES"
705 FORH=1TON
710 INPUTA$(H)
715 NEXTH
720 RETURN
```

Gráficos en Tres Dimensiones

Para VIC-20 con Superexpander o Graphics hace mucho tiempo que los que hacemos Club Commodore andábamos como locos persiguiendo un programa que permita representar en pantalla gráficos en tres dimensiones. Gracias a la colaboración que hoy publicamos desaparece una de nuestras mayores obsesiones, a ver si alguien se anima y lo adapta para el Commodore 64.

Juan Santa María Fernández (18 años)
C/ Maestro Pérez Cabrero, 4
Tel.: 201 72 90. Barcelona-21
Javier Anglada Cantarell
C/ Mandri, 68
Tel.: 247 91 62. Barcelona-22

Presentamos un programa de gráficos que representa la superficie gráfica generada por una función z=f(x,y) y que da una visión tridimensional de la misma.

Nosotros creemos oportuno mandarlo ya que hasta ahora los programas de gráficos que han salido en esta revista sólo trataban de la representación unidimensional de una función. Nuestro programa creemos que abre una puerta a las gráficas de representación de superficies.

Aunque este programa no es del todo completo y se puede prestar a muchas modificaciones, según la memoria que cada uno tenga, creemos que ofrece una visión básica del modo de programar superficies a un nivel adsequible para todos los niveles.

Quizás una de las mayores dificultades que hemos tenido que salvar era la de tapar las partes ocultas ya que al principio quedaban las gráficas poco definidas por esta causa. Una vez salvada esta primera dificultad nos dedicamos a proporcionar la altura del plano.

El programa en sí tiene cinco partes bien diferenciadas que paso a explicar brevemente:

Líneas 100-200: Introducción de variables; dimensión de puntos máximos y mínimos para las partes ocultas y preparación del ordenador al modo gráfico.

Líneas 210-260 y 470-510: Calcula en primer lugar X2 e Y2, estos valores los gira dando la X e Y; a partir de estas calcula D y por fin calcula Z=f(x,y)

Líneas 270-340: Mira si ha de dibujar el siguiente punto y calcula su altura real en la pantalla.

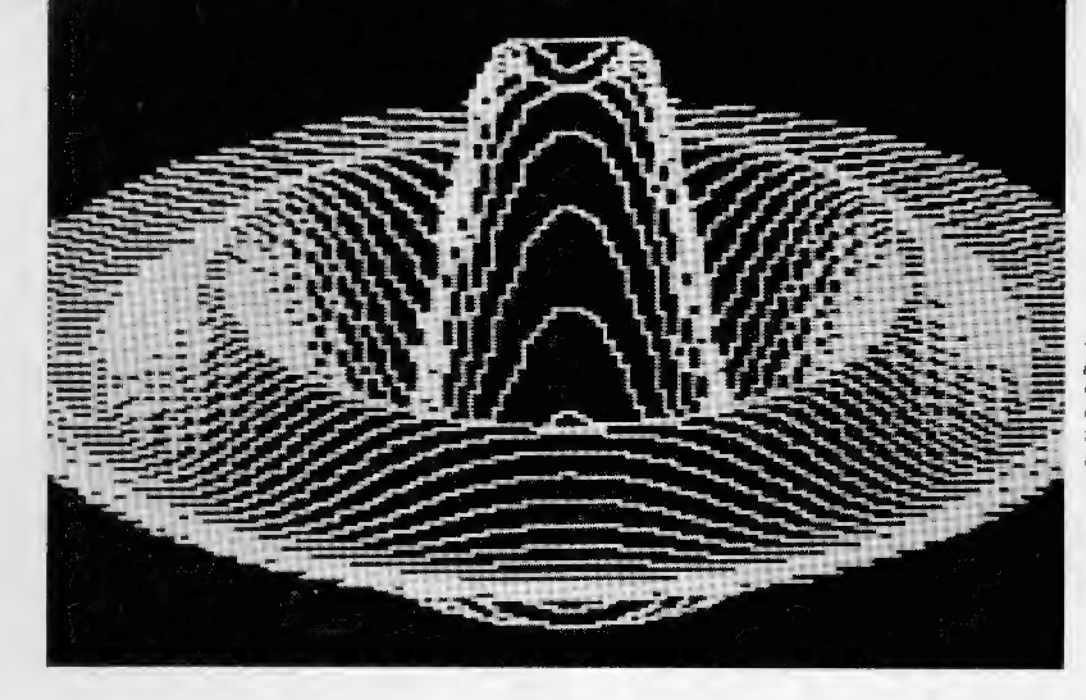
Líneas 350-370: Dibuja la línea y actualiza máximos y mínimos.

Líneas 410-460: Calcula el último punto que es oculto o el último punto visto y lo sitúa en un máximo o en un mínimo según convenga.

Como he venido observando que todos los programas son para utilizar el SUPEREXPANDER y que los usuarios de GRAPHVICS (un casette para gráficos que vende la compañía Indescomp) están un poco discriminados, me he tomado la molestia de señalar las diferencias entre los dos sistemas. Así pues los usuarios de este último sistema sólo tienen que cambiar las siguientes líneas:

```
180 DIMM(1,151)
190 FOR X=0 TO 151: M(0,X)=159:
    M(1,X)=0: NEXT.
200 &S9,2
```





Aspecto de la pantalla del VIC con el programa del artículo. (Foto de los autores).

```
210 FOR J=0 TO 159 STEP 3
220 Y2=Y0+(Y1-Y0)*J/159
240 FOR I=0 TO 151
250 X2=X0+(X1-X0)*1/151
270 S=(Z*Z0+J)
280 IF S   159 THEN S=159
350 & L(I-1), SI,I,S
```

Una vez cargado el programa y resueltas las modificaciones lo que tenéis que hacer para que funcione es lo siguiente:

En primer lugar introducid la función ha ejecutar en la línea 500.

Seguidamente hacéis un RUN y se introducen las variables que ahora os explico.

Los valores máximos y mínimos de X e Y es el intervalo que queréis que represente de los ejes X e Y.

La altura es el factor por el que multiplicamos la altura real de los puntos y que tendrán estos en la pantalla.

En intervalo se introduce la distancia al centro que queréis que se vea en la pantalla según X e Y.

El giro se introduce en radianes, pero si queréis introducirlo en grados sólo tenéis que cambiar la línea 170 y poner en ella:

170 INPUT "GIRO":";AL:AL=
=180*AL/π

Por último os damos un ejemplo para que lo probéis y que os dará la figura que mandamos:

En la línea 500 Z=EXP(-D*D)*SIN (2*π*D) que es la función que ponemos en el programa.

Poner en X máxima y mínima y en Y máxima y mínima 2,-2.

En altura para el SUPEREXPANDER poner 500 y para el GRAPHVICS poner 30.

En intervalo ponéis 2 y por último en giro da lo mismo lo que pongáis ya que la gráfica es totalmente simétrica respecto a Z.

Si ponéis los datos como os hemos dicho veréis aparecer poco a poco la gráfica de la fotografía en vuestra pantalla, ahora sí, es importante que recordéis que el programa va muy lento y que por lo tanto tendréis que tener un poco de paciencia hasta que lo termine, más o menos suele acabarlo en unos 20 minutos según la función y los puntos a representar en la misma.

Por último y para terminar hemos de agradecer la colaboración de la escuela de informática TRONIK en Barcelona.

Esperamos que os guste el programa.

Para el apartado de MAGIA he observado que con el cartucho AYUDA AL PROGRAMADOR se pueden listar los programas de abajo a arriba para ello al hacer LIST pulsar las teclas CTRL y Q y el programa aparecerá lentamente al llegar al final del mismo si pulsais las teclas CTRL y A el programa aparecerá por bloques de abajo a arriba. Hay veces que esto no ocurre, si pasa, volver a escribir List y volver a pulsar la tecla Q.

```
PROGRAMA: GRAFICOS EN TRES DIMENSIONES


100 REM "[1SHIFG]RAFICOS[1SPC]3D
."
110 PRINTCHR$(14)
120 CLR
130 INPUT"X[1SPC]MAX.,MIN:";X1,X
0
140 INPUT"Y[1SPC]MAX.,MIN:";Y1,Y
0
150 INPUT"FACTOR[1SPC]Z:";Z0
160 INPUT"INTR.:";I0
170 INPUT"GIRO:";AL
180 DIMM(1,200)
190 FORX=0TO200:M(0,X)=1023:M(1,
X)=0:NEXT
200 GRAPHIC2:COLOR0,0,1,0
210 FORJ=0TO800STEP12
220 Y2=Y0+(Y1-Y0)*J/800
230 G=1
240 FORI=1TO170
250 X2=X0+(X1-X0)*I/170
260 GOSUB470
270 S=900-(Z*Z0+J)
280 IFS>=1023THENS=1022
290 IFS<0THENS=0
300 IFD>I0THENG=1:GOTO380
310 IFG=1THENG=0:GOTO360
320 IFS>M(0,I)ANDS<M(1,I)ANDR=0T
HENR=1:GOTO440
330 IFS>M(0,I)ANDS<M(1,I)THENGOT
O390
340 IFR=1THEN410
350 DRAW2,(I-1)*6,SITOI*6,S
360 IFS>M(1,I)THENM(1,I)=S
370 IFS<M(0,I)THENM(0,I)=S
380 SI=S
390 NEXT:NEXT
400 END
410 IFS>=M(1,I)THENSI=M(1,I-1)
420 IFS=<M(0,I)THENSI=M(0,I-1)
430 R=0:GOTO350
440 IFSI<=M(0,I-1)THENS=M(0,I):G
OTO350
450 IFSI>=M(1,I-1)THENS=M(1,I)
460 GOTO350
470 X=X2*SIN(AL)+Y2*COS(AL)
480 Y=Y2*SIN(AL)-X2*COS(AL)
490 D=SQR(X*X+Y*Y)
500 Z=EXP(-D*D)*SIN(2*π*D)
510 RETURN
```

2 PROGRAMAS

Sala y Diccionario

Para VIC-20 sin ampliación (la capacidad del programa diccionario se puede aumentar con ampliación) Gorka Bueno nos manda dos programas interesantes; esperamos que los usuarios de su generación tomen ejemplo y nos manden colaboraciones.

Gorka Bueno Mendieta (13 años)
Part. de Alzola; 3-3º G
Bilbao-12
Tel.: 443 28 80

Hola. Me llamo Gorka Bueno, tengo 13 años, vivo en Bilbao, y os mando dos colaboraciones que espero os gusten. El primer programa, "SALA", es un simple juego de habilidad, en el que tienes que pasar por la sala sin chocar con los verdugos. Cabe en la memoria sin necesidad de ampliación. El segundo es un diccionario.

Si no se es poseedor de ampliación de memoria no introduzcáis las sentencias REM, debido a que ocupan mucha memoria no introduzcais las sentencias ria escribid las sentencias sin el REM. El programa en si tiene cinco posibilidades:

1.—Meter la palabra en un idioma (castellano), y sacar su equivalente en otro, si es que existe (euskera).
2.—De forma inversa a la anterior.
3.—Listar todas las palabras en los dos idiomas. Al terminar pulsar "ASTERISCO".
4.—El ordenador elige una palabra al azar en un idioma (castellano), y tú has de responder en el otro.
5.—Funcionamiento inverso al anterior.

Las posibilidades cuatro y cinco sólo son posibles con ampliación, debido a la gran cantidad de DIM.

Yo he puesto como ejemplo el castellano y el euskera como idiomas, pero se pueden cambiar por otros.

El gran problema es hacer las datas, pero con un poco de paciencia...



```
PROGRAMA: SALA
1 CO=30720:Q=7943:POKE36878,15:R
ESTORE
2 PU=0:POKE36879,141
5 GOSUB900
6 DIMD(16):DIMO(16)
10 PRINT"[1HOME][1CRSRD][1CRSRR]
[9SPC][2CRSRR][9SPC]";
12 TI$="000000"
17 Q=7943:PRINT"[1CLR][22COMM+]"
;
20 PRINT"[1COMM+][9SHFSPC][2COMM
+][9SHFSPC][1COMM+]";
30 PRINT"[22COMM+]";
40 PRINT"[10COMM+][11SHFSPC][1CO
MM+]";
50 PRINT"[10COMM+][1SHFSPC][9COM
M+][1SHFSPC][1COMM+]";
60 PRINT"[3COMM+][13SHFSPC][2SPC
][1SHFSPC][1COMM+][1SHFSPC][1COM
M+]";
70 PRINT"[3COMM+][13SHFSPC][2SPC
][1SHFSPC][1COMM+][1SHFSPC][1COM
M+]";
80 PRINT"[3COMM+][13SHFSPC][2SPC
][1SHFSPC][1COMM+][1SHFSPC][1COM
M+]";
90 PRINT"[3COMM+][13SHFSPC][2SPC
][1SHFSPC][1COMM+][1SHFSPC][1COM
M+]";
100 PRINT"[3COMM+][13SHFSPC][2SP
C][1SHFSPC][1COMM+][1SHFSPC][1CO
MM+]";
110 PRINT"[3COMM+][13SHFSPC][2SP
C][1SHFSPC][1COMM+][1SHFSPC][1CO
MM+]";
120 PRINT"[2SHFSPC][1COMM+][16SH
FSPC][1COMM+][2SHFSPC]";
130 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
140 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
150 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
160 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
170 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
180 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
190 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
200 PRINT"[1COMM+][1SHFSPC][1COM
M+][16SHFSPC][3COMM+]";
210 PRINT"[1COMM+][1SHFSPC][8COM
M+][1SHFSPC][11COMM+]";
220 PRINT"[1COMM+][10SHFSPC][11C
OMM+]";
230 PRINT"[21COMM+]";
235 POKE8164+22-1,102:POKE8164+2
2-1+30720,6
250 REMDIMX$(5):FORI=0TO14:X$(I)
=CHR$(I+65):NEXT
280 FORY=1TO15
290 H=INT(RND(1)*15)+1:T=INT(RND
(1)*15)+1
300 D(Y)=7680+88+(22*Y)+H+3:O(Y)
=7680+88+(22*Y)+T+3
310 POKED(Y),81:POKEO(Y),81:POKE
D(Y)+CO,7:POKEO(Y)+CO,5
320 NEXT
330 FORY=1TO15
335 W=0
340 GETE$
345 IFE$=""THEN410
350 READW
390 POKE36876,200:FORA=1TO25:NEX
T:POKE36876,0
400 IFPEEK(Q+W)=81THEN1000
410 POKEQ,32:Q=Q+W:POKEQ,90:POKE
Q+CO,6:W=0
412 IFQ=7922THENRESTORE:PU=PU+(2
6-VAL(TI$))*100:GOTO10
413 PRINT"[1HOME][1CRSRD][11CRSR
R][8SPC]"
414 PRINT"[1HOME][1CRSRD][1CRSRR
]"PU:PRINT"[1HOME][1CRSRD][11CRS
RR]"26-VAL(TI$):IFTI$="000026"TH
EN2000
415 POKED(Y),32:POKEO(Y),32
416 D(Y)=D(Y)+1:O(Y)=O(Y)+1:IFPE
EK(D(Y))=90ORPEEK(O(Y))=90THEN10
00
417 IFD(Y)=7680+88+(22*Y)+19THEN
D(Y)=D(Y)-16
418 POKED(Y),81:POKED(Y)+CO,7
420 IFO(Y)=7680+88+(22*Y)+19THEN
O(Y)=O(Y)-16
430 POKEO(Y),81:POKEO(Y)+CO,5
470 NEXT
480 GOTO330
500 DATA-1,-22,-22,-22,-22,-22,-
22,-22,-22,-1,-1,-1,-1,-1,-1,-1,
-1,-1,-1,22,22,22,22
510 DATA22,22,22,22,22,22,22,22,
22,22,22,22,22,22,-1,-1,-1,-1,-1
,-1,-1,-1,-1
520 DATA-22,-22,-22,-22,-22,-22,
-22,-22,-22,-22,-1
900 PRINT"[1CLR]**[1SPC]SALA[1SP
C]DE[1SPC]TORTURA[1SPC]***"
910 PRINT"[5CRSRD][1SPC]DEBES[1S
PC]PASAR[1SPC]POR[1SPC]LA[3SPC]"

920 PRINT"[1SPC]SALA[1SPC]SIN[1S
PC]CHOCAR[1SPC]CON[2SPC]"
930 PRINT"[1SPC]LOS[1SPC]VERDUGO
S.[1SPC]PARA[3SPC]"
940 PRINT"[1SPC]MOVERTE,[1SPC]TI
ENES[1SPC]QUE[2SPC]"
950 PRINT"[1SPC]PULSAR[1SPC]UNA[
1SPC]TECLA[5SPC]"
955 PRINT"[4SPC]CUALQUIERA"
960 PRINT"[3CRSRD][6CRSRR]PULSA[
1SPC]'C'"
970 GETA$:IFA$=""THEN970
980 IFA$="C"THENRETURN
990 GOTO970
1000 PRINT"[1HOME][9CRSRD][3CRSR
R]TE[1SPC]HA[1SPC]APRESADO"
1010 PRINT"[6CRSRD][6CRSRR]OTRA[
1SPC]VEZ?"
```



```
1020 GETA$:IFA$="S"THENRUN
1030 IFA$="N"THENPRINT"[1CLR]":E
ND
1040 GOTO1020
2000 PRINT"[1HOME][9CRSRD][3CRSR
R]TIEMPO[1SPC]ACABADO"
```

```
2010 PRINT"[6CRSRD][6CRSRR]OTRA[
1SPC]VEZ?"
2020 GETA$:IFA$="S"THENRUN
2030 IFA$="N"THENPRINT"[1CLR]":E
ND
2040 GOTO2020
```

PROGRAMA: DICCIONARIO

```
1 RESTORE
2 REMGOSUB300
10 RESTORE
11 PRINT"[1CLR]****[1SPC]DICCION
ARIO[1SPC]*****"
20 PRINT"[3CRSRD]1)[1SPC]BUSCAR[
1SPC]EN[1SPC]EUSKERA"
30 PRINT"[1CRSRD]2)[1SPC]EN[1SPC
]CASTELLANO"
40 PRINT"[1CRSRD]3)[1SPC]LISTAR"

42 REMPRINT"[1CRSRD]4)[1SPC]PREG
UNTAR[1SPC]EUSKERA"
43 REMPRINT"[1CRSRD]5)[1SPC]EN[1
SPC]CASTELLANO"
45 PRINT"[4CRSRD]"
50 INPUTS
60 IFS<1ORS>5THEN10
70 IFS=2THEN150
80 IFS=3THEN220
83 REMIFS=4THEN340
84 REMIFS=5THEN400
90 INPUT"[1CLR][2CRSRD]";A$
100 READB$:READC$
105 IFB$="NO[1SPC]EXISTE[1SPC]EN
"THENPRINTB$;"[1SPC]";C$:GOSUB99
0:GOTO10
110 IFA$<>B$THEN100
120 D$=C$
130 PRINT"[2CRSRD]"A$,D$
140 GOSUB990:GOTO10
150 INPUT"[1CLR][2CRSRD]";A$
160 READB$:READC$
170 IFB$="NO[1SPC]EXISTE[1SPC]EN
"THENPRINTB$;"[1SPC]";C$:GOSUB99
0:GOTO10
180 IFA$<>C$THEN160
190 D$=B$
200 PRINT"[2CRSRD]"A$,D$
210 GOSUB990:GOTO10
220 PRINT"[1CLR]"
230 READA$:READB$
240 IFA$="NO[1SPC]EXISTE[1SPC]EN
"THENPRINT"FIN":GOSUB992:GOTO10
250 PRINTA$,B$
260 GOSUB990:GOTO230
300 REMPRINT"[1CLR][9CRSRD][2CRS
RR]ESPERA,[1SPC]POR[1SPC]FAVOR"
301 REMDIMX$(500):DIMY$(500)
305 REMX=0:Y=0:L=-2
```

```
307 REMREADA$:READB$:L=L+2
310 REMIFA$="NO[1SPC]EXISTE[1SPC
]EN"THENRETURN
320 REMX$(X)=A$:Y$(Y)=B$
330 REMX=X+1:Y=Y+1:GOTO307
340 REMPRINT"[1CLR]"
350 REMH=INT(RND(1)*(L/2))+1
360 REME$=X$(H)
370 REMPRINT"[3CRSRD]"E$:INPUT"[
1CRSRD]";W$
380 REMIFW$<>Y$(H)THENPRINT"[3CR
SRD][1CRSRR]LO[1SPC]SIENTO,[1SPC
]ESTA[1SPC]MAL":PRINT"[1CRSRD][1
CRSRR]ES:";Y$(H):GOSUB990:GOTO10

390 REMPRINT"[4CRSRD][5CRSRR]O.K
.":GOSUB990:GOTO10
400 REMPRINT"[1CLR]"
410 REMH=INT(RND(1)*(L/2))+1
420 REME$=Y$(H)
430 REMPRINT"[3CRSRD]"E$:INPUT"[
1CRSRD]";W$
440 REMIFW$<>X$(H)THENPRINT"[3CR
SRD][1CRSRR]LO[1SPC]SIENTO,[1SPC
]ESTA[1SPC]MAL":PRINT"[1CRSRD][1
CRSRR]ES:"X$(H):GOSUB990:GOTO10
450 REMPRINT"[4CRSRD][5CRSRR]O.K
.":GOSUB990:GOTO10
990 GETA$:IFA$=""THEN990
991 RETURN
992 GETA$:IFA$<>"*"THEN992
993 RETURN
1180 DATAENERO,URTARRILA,FEBRERO
,OTSAILA,MARZO,MARTXOA,ABRIL,API
RILA,MAYO,MAIATZA
1190 DATAJUNIO,EKAINA,JULIO,UZTA
ILA,AGOSTO,ABUZTUA,SEPTIEMBRE,IR
AILA,OCTUBRE,URRIA
1200 DATANOVIEMBRE,AZAROA,DICIEM
BRE,ABENDUA
1210 DATALUNES,ASTELEHENA,MARTES
,ASTEARTEA
1220 DATAMIERCOLES,ASTEASKENA,JU
EVES,OSTEGUNA,VIERNES,OSTIRALA,S
ABADO,LARUMBATA
1230 DATADOMINGO,IGANDEA,PRIMAVE
RA,UDABERRIA,VERANO,UDA,OTO@O,UD
AZKENA,INVIERNO,NEGUA
10000 DATANO EXISTE EN,EL DICCIO
NARIO
```

Cuando se es COMMODORE es muy difícil ser modesto

COMMODORE 64

Cuando se tiene 64 K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con sprites, un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos, la más completa gama de programas educativos, profesionales y de videojuegos...; en resumen, cuando se es un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, es muy difícil decir sin orgullo que eres un Commodore-64.

Claro que más difícil todavía es decir sin orgullo que tienes un Commodore-64. ¿Por qué no lo comprueba?

COMMODORE 64 LE DA ACCESO A MUCHOS ACCESORIOS

Unidad simple de disco (Monofloppy) 170 K.

Cassette.

Plotter e impresora, 4 colores, 14 c.p.s.

Impresora matricial, tractor, 30 c.p.s.

Interface RS232.

Joy Stick.

Paddle.

Cursos de Introducción al BASIC.



COMMODORE 64 LE MUESTRA PARTE DE SUS PROGRAMAS

Utilitarios y lenguajes

MONITOR LENGUAJE MAQUINA.
FORTH.
LOGO.
PILOT.
MACRO ASSEMBLER.
PROGRAMMER'S UTILITIES.
TURTLE GRAPHICS II.
MASTER.

Sistemas operativos

FILE/BOSS.
CP/M.

Programas de aplicaciones

EASY SCRIPT.
Proceso de texto de gran potencia.

CALC RESULT.
Hoja electrónica de cálculo.

EASY CALC RESULT.
Versión simplificada del CALC RESULT.

MAGIC DESK.
Proceso de texto y gestión de ficheros.

AGENDA TELEFONICA.

Programas educativos

MUSIC MACHINE.
MUSIC COMPOSER.
VISIBLE SOLAR SYSTEM.
SPEED/BINGO MATH.
FISICA I.
MATEMATICAS I.
HISTORIA I.
GEOGRAFIA I.
GEOGRAFIA II.
JUEGOS EDUCATIVOS.
TEMAS MONOGRAFICOS.
CONOCIMIENTOS GENERALES.
QUIMICA I.

Juegos

JUPITER LANDER.
KICKMAN.
SEAWOLF.
RADAR RAT RACE.
TOOTH INVADERS.
LAZARIAN.
OMEGA RACE.
LE MANS.
PINBALL SPECTACULAR.
AVENGER.
SUPERMASH.
FROGMASTER.
GRID RUNNER.
ATTACK OF THE MUTANT CAMELS.
THE PIT.
MR. TNT.
6 GAME PROGRAMS.
BINGO.
ROOTING TOOTING.
MINESSOTA FAT'S POOL CHALLENGE.

... y seguimos ampliando la lista

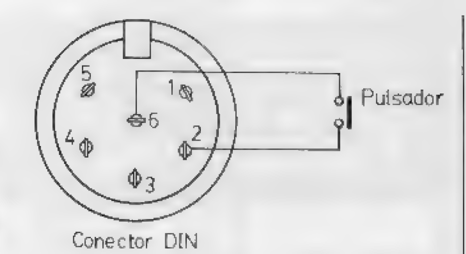
El ordenador personal de la familia más potente



MICROELECTRONICA Y CONTROL, S.A.
c/. Taquígrafo Serra, 7, 5.º. Barcelona-29
c./ Princesa, 47, 3.º G. Madrid-8



Comando Old



Macho visto desde las soldaduras

Por Diego Romero

¿Un programa que no ocupa memoria?...

Quizás en alguna ocasión has sufrido la amarga experiencia de perder un programa que tardaste varias horas en teclear, y por alguna extraña circunstancia el ordenador se quedó "colgado" sin hacer caso de las teclas RUN/STOP y RESTORE. En este caso, la única solución posible era apagarlo...

Si te ocurrió y no quieres que vuelva a suceder, o si eres de los que se aplican aquello de "Hombre prevenido...", este artículo puede interesarte.

¿Por qué no respondía a STOP y RESTORE?

Ni el VIC-20 ni el C-64 disponen de una tecla de RESET que es la que podría haber sacado al ordenador de aquella situación.

La tecla de "RESTORE" es realmente una tecla como otra cualquiera del teclado, la diferencia es que cuando se la pulsa junta con la de RUN/STOP, el ordenador ejecuta una rutina que restaura las condiciones normales; pero si algún diabólico POKE del programa afectó a la rutina que comprueba esta tecla, tu ordenador no responderá al pulsarla.

¿Cómo añadir una tecla "RESET"?

Para incorporar una tecla de RESET auténtica a tu ordenador, tienes tres lugares posibles donde conectarla:

El Port de Usuario, el Port de Ampliación y el Bus de Serie.

Realmente consiste en conectar un pulsador entre la línea de "reset" de la CPU y masa. El lugar más sencillo es el Bus de Serie, pero si prefieres puedes conectarlo en cualquiera de los otros dos (mira el manual de usuario de tu VIC o C-64).

¿Cómo hacerlo?

Necesitarás disponer de un soldador o de un amigo que lo tenga y te ayude. Compra un conector macho DIN de 270 grados con 6 patillas como el del dibujo, y un pulsador miniatura que quepa en la parte trasera del conector (para poder fijarlo mecánicamente a éste).

Suelda el pulsador entre las patillas 2 y 6 del conector.

Para probarlo conéctalo al conector trasero de tu VIC o C-64. Si tienes la unidad de disco incorporada, puedes conectar el pulsador en la parte trasera de ésta.

¡CUIDADO! Si tienes la unidad de disco, debes comprobar que no hay ningún disco dentro, antes de pulsar el "RESET", pues podría borrar algún programa o dato.

Cuando tengas todo conectado puedes encender el ordenador. Después teclea algunas cosas en la pantalla, pulsa el nuevo "RESET" y, si todo va bien, desaparecerá lo que había en la pantalla y aparecerá el mensaje de puesta en marcha, como si hubieses apagado y encendido el ordenador. Si no es así, repasa las soldaduras, comprueba que son las patillas 2 y 6, y vuelve a probarlo.

Una vez que el Hardware va bien, vamos al Software...

Teclea el listado que aparece a continuación, guarda una copia en cinta para utilizarlo otras veces, puedes llamarlo "generador de OLD" para no confundirlo con el programa OLD.

Pon otra cinta en el DATASSETTE y teclea RUN con RETURN. Aparecerá el mensaje "Press Record & Play", hazlo y espera el READY... Ya dispones de una copia del programa-comando OLD en cinta. Si te apareció el mensaje "ERROR EN DATOS", revisa las líneas con sentencias DATA y corrígelas.

¡OJO!... El programa que aparece en el listado es el que genera el programa OLD, pero no es el OLD, ya que éste último no se puede listar al no ocupar ni un solo BYTE de la memoria reservada para programas.

¿Cómo utilizar el programa OLD?

Con la cinta que has preparado, puedes hacer la prueba:

1. Teclea un programa corto o carga cualquiera.
2. Teclea NEW y pulsa RETURN o el pulsador "RESET" que has preparado.
3. Comprueba que el programa ya no está (LIST).
4. Teclea LOAD "OLD" y pulsa RETURN. Pon la cinta OLD y espera.
5. Aparecerá el mensaje FOUND OLD y después alguna "basura" en la pantalla. ¡CUIDADO! NO PULSES RETURN AHORA.
6. Pulsa RUN/STOP y RESTORE.
7. Teclea LIST y pulsa RETURN... OK... Programa recuperado...

¿Cómo fue?

El programa que aparece en el listado coloca en las direcciones $1C00 y siguientes una rutina de lenguaje máquina que genera un programa en cinta que se llama OLD.

Cuando tecleas LOAD "OLD" cargas un programa de solo dos bytes que interceptan el sistema operativo del ordenador, y le hacen que ejecute automáticamente una rutina en código máquina, que está situada precisamente bajo el nombre del programa OLD, por eso no ocupa un solo byte de la memoria de programa.

¿Parece complicado?... Lo es, si no conoces tu VIC ó C-64 ni su lenguaje máquina, pero es sencillo si sabes código máquina y analizas la rutina que coloca el "generador de OLD" en $1C00.

Si no sabes lenguaje máquina, puedes seguir el cursillo que paso a paso comienza en este mismo número (pág. 62), así como coleccionar las recetas del "Recetario de POKES" (Pág. 39) y más adelante intentas descifrarlo y/o ya te lo explicaré más detalladamente.

Un saludo...

```
10 POKE56,28:POKE55,0:CLR:AD=28*256
20 READ[illegible]
30 [illegible]
40 IF[illegible]THENPRINT"ERROR[SPC]EN[SPC]D
ATOS":END
50 PRINT"PRESS[SPC]RECORD[SPC]&[SPC]P
LAY":PRINT"[SPC]Y[SPC]PULSA[SPC]UNA[SPC]T
ECLA"
60 GET A$:IF A$=""THEN60
80 SYS28*256:END
100 DATA 169, 0, 32, 144, 255
110 DATA 162, 81, 160, 3, 142, 38, 3
 140
120 DATA 39, 3, 169, 1, 162, 1, 160,
 1
130 DATA 32, 186, 255, 169, [illegible], 16[illegible]
 54, 160
140 DATA 28, 32, 189, 255, 162, [illegible]
160, 3
150 DATA 134, 254, 13[illegible], [illegible], 1[illegible]
4, 162, 40
160 DATA 160, [illegible], [illegible], 210, 255, 32, 1
38, 255
170 DATA 96, 79, 76, 68, 32, 32, 32,
 32
180 DATA [illegible], 32, 32, 32, 32, 32, 3[illegible]
 32
190 DATA 32, 166, 43, 164, 44, 134
254, 13[illegible]
200 DATA 255, 160, 4, 177, 254, 240,
 [illegible], 200
210 DATA 208, 249, 200, 15, 1[illegible], [illegible]
 24, 101
220 DATA 254, 145, 254, 200, 1[illegible], 0,
 101, 255
230 DATA 145, 254, 169, 0, 1[illegible], 254,
 20[illegible], 30
240 DATA 200, 177, 254, 208, 25, 16[illegible]
, 2, 24
250 DATA 101, 254, 133, 45, 1[illegible], 4[illegible],
 13[illegible], 49
260 DATA 169, 0, 101, 255, 13[illegible], 46
133, 48
270 DATA 133, 50, [illegible], 13[illegible], [illegible], 9[illegible]
160, 0
280 DATA 17[illegible], 254, 170, 200, 177, [illegible]
4, 133, 255
290 DATA 134, 254, 56, 176, 205, 1315
3
```

VIC-20
C-64



El Misterio de los Pokes

Por Diego Romero Sabio

Este mes comenzaremos por las direcciones 0, 1 y 2 del VIC-20, y las que son funcionalmente equivalentes en el C-64, es decir, las direcciones 784, 785 y 786.

El manual de usuario no deja bien claro cómo se puede utilizar la función URS (X) de nuestro BASIC, por tanto vamos a tratar de aclarar un poco las dudas que pudieran surgir al respecto.

Los POKEs en las tres direcciones anteriores son realmente una instrucción de salto en código máquina a la rutina que queremos que ejecute el ordenador cada vez que se encuentre una función BASIC de este tipo. El salto propiamente dicho es el valor $4C (76) que ocupa la primera de las posiciones, la siguiente posición es la parte baja de la dirección, y la última es la parte más alta de ésta. La X en la función USR (X) es el argumento de la función y es transferido por el intérprete BASIC al acumulador de coma flotante número 1 (de éste ya hablaremos más adelante en el cursillo de lenguaje máquina). Este acumulador será estudiado más adelante en el cursillo de lenguaje máquina, de momento sólo debemos saber que el argumento de la función USR es lo que va entre paréntesis y que es transferido a la rutina de código máquina. La función nos retornará un valor que será la respuesta de la rutina que hayamos ejecutado, y podemos imprimirlo con PRINT USR (X).

Probar el siguiente ejemplo:

```
10  POKEO,76:POKE1,204:POKE2,220:
    REM ESTA LINEA ES PARA EL
    VIC-20
20  INPUT "DATO";DA
30  PRINT INT (DA), USR(DA): GOTO20
```

Para el C-64 cambiar la línea 10 por la siguiente:

```
10  POKE784,76:POKE785,204:POKE786,
    188:REM ESTA LINEA ES PARA
    EL C-64
```

El resto del programita de demostración es igual al del VIC-20.

Cuando programemos en código máquina, podremos realizar funciones especiales definidas por nosotros mismos y transferir los parámetros desde BASIC gracias a la función USR(X).

En este mismo número comenzamos un cursillo de lenguaje máquina para los que quieran aprender...

El misterioso POKE que hemos realizado en este programa, lo que hace realmente es poner la instrucción de salto a la rutina que ejecuta la función INT(X) en el intérprete BASIC.

Hasta ahora hemos descubierto el "Misterio de tres POKES" y la inexplicable función USR(X) que aparece en el manual de usuario.

En el C-64, el manual de usuario está equivocado, pues si lo miráis, veréis que dice que son las direcciones 0, 1 y 2, sin embargo, la guía de referencia del programador sí pone las direcciones correctas.

En el próximo número continuaremos describiendo la función de los POKES a distintas posiciones de memoria.

Si tenéis dudas, hacerme llegar cartas a la revista e intentaré resolverlas.

Hasta pronto...

GLOSARIO

Address
Dirección.—Número que identifica una posición determinada de memoria que puede ser leída o escrita.

Binario
Sistema de numeración con base 2. Utiliza los guarismos 0 y 1.

BIT
BInary digiT.—Dígito binario. Unidad elemental de información en el sistema binario de numeración. Puede tomar los valores 0 ó 1.

Buffer
Memoria intermedia utilizada para la comunicación entre periféricos o entre estos y la CPU (ej. el de cassette o el de teclado).

Byte
Octeto.—Grupo de ocho bits que pueden ser leídos o almacenados por la CPU en cualquier posición de memoria.

GLOSARIO

CPU
Central Processing Unit. Unidad central de proceso. Es la encargada de interpretar y ejecutar las instrucciones del programa.

Decimal
Sistema de numeración con base 10. Utiliza los guarismos del 0 al 9.

Hexadecimal
Sistema de numeración con base 16 que utiliza los guarismos del 0 al 9 y de la A a la F.

Octal
Sistema de numeración con base 8 que utiliza los guarismos del 0 al 7.

Reset
Señal o procedimiento de puesta a cero o inicialización de los circuitos y programas.



RINCON DEL 700

Serie 700 (5): UTILITARIOS

Jordi Sastre

¡Cuántas veces el programador escribe un mismo programa para realizar una misma función!: mirar un bloque de disco, examinar un fichero desde determinado registro, la típica rutina para cambiar el número de periférico o el nombre del disco, etc... Es muy útil para los programadores disponer de pequeños programas que realicen funciones concretas en un momento dado. A veces para integrarlos en una aplicación, pero casi siempre para usarlos como herramientas propias de trabajo. A estos programas se les llama utilitarios, o programas de utilidad.

Hay secuencias de instrucciones que han hecho historia por haberse repetido cientos de veces. Por ejemplo, es fácil visualizar el contenido de un bloque de disco mediante el siguiente programa:

```
10 OPEN 15,8,15 : OPEN 2,8,2"#"
20 PRINT#15,"U1";2;0;P;S: REM P=
   PISTA S=SECTOR
30 FORX=1TO255 : GET#2,A$ :
   PRINTA$; : NEXT
40 CLOSE2 : CLOSE15
```

En base a esto, el programa suele ir modificándose para ceñirse a unos bytes más concretos, visualizar el valor ASCII en lugar del carácter, modificar bytes del bloque sin afectar a los demás, perseguir los encadenamientos entre bloques de ficheros, etc... Cuando terminamos lo que queríamos hacer, el programa que tenemos no se parece en nada al original después de las sucesivas variaciones. Hemos conseguido nuestro propósito de la forma más manual o artesana que existe. Es muy importante que el programador tenga los conocimientos necesarios para acceder de esta manera al disco, precisamente por ser la más rudimentaria; de la misma manera que a pesar de disponer de calculadora debemos ser capaces de sumar o restar sin ella: la calculadora la reservamos para operaciones más difíciles.

MEC.SOFT está desarrollando utilitarios para la serie 700. Ya se comercializa el primer lote de los mismos, compuestos por los siguientes programas.



MD/CREA.PANT700
Creación de Pantallas MEC/DOS.

MD/EXPA.REL700
Expansión de Ficheros Relativos MEC/DOS.

COPYFILE
Copia de ficheros entre periféricos.

LOOKDIR
Visualización especial del directorio.

DISKMON
Monitor de bloques de disco.

UNIT CHANGE
Cambio de número de periférico.

HEADER CHANGE
Cambio del nombre del disco.

FORMAT8000
Cambio de formato de programas 700 a 8000

Entre paréntesis, aprovecho para decir que MEC.SOFT aceptará con muchísimo gusto ideas para nuevos utilitarios.

Este grupo es bastante heterogéneo en cuanto a tipos de utilitarios. Los primeros, caracterizados por su prefijo MD/, son utilitarios de trabajo para con MEC/DOS: MD/CREA.-PANT700 confiere gran potencia a la hora de diseñar los formatos de pantalla, y MD/EXPA.REL700 permite solucionar lo que siempre se prevé pero siempre sucede: que un fichero se quede pequeño. Los seis restantes afectan únicamente a la unidad de discos. Para copiar programas o ficheros de un periférico a otro, la única solución era leer secuencialmente (GET#) los datos del fichero abierto en el periférico origen para grabarlos en otro fichero abierto en el periférico destino, mediante un corto programa en Basic (el comando COPY no copia entre diferentes periféricos). COPYFILE hace exactamente esto, pero en código máquina (sinónimo de rapidez). Por otra parte, LOOKDIR y DISKMON hacen lo que muchos programadores hacían manualmente con pequeños programas que siempre se escribían de nuevo porque nunca había tiempo para depurarlos y grabarlos: el primero visualiza lo que los comandos CATALOG y DIRECTORY no visualizan (el track y sector donde empieza un fichero, y la longitud del record de los ficheros relativos,... etc.), mientras que DISKMON edita el contenido de un bloque de disco para visualizar o modificar bytes del mismo. Todo programador tiene ya su rutina para cambiar el número de un periférico o el nombre de un disco; no tendría ningún sentido que MEC.SOFT hubiera confeccionado los utilitarios UNIT CHANGE y HEADER CHANGE, si no fuera porque cada día hay nuevos programadores que se incorporan a la familia Commodore. Por último, FORMAT8000 es un tema nuevo en utilitarios. Hasta ahora los programas



eran siempre compatibles entre los ordenadores de sistemas existentes (CBM-3032, CBM-4032 y CBM-8032), pero la serie 700 carga los programas Basic en distinta zona de memoria, lo que incompatibiliza, de momento, la carga de programas entre estas CPUs.

Todos los utilitarios están escritos en Basic, en ocasiones apoyado por rutinas en código máquina, y siempre utilizando comandos MEC/DOS, por lo que es necesario tener conectado el cartucho.

Dedicaré una serie de artículos a comentarlos: qué hacen, cómo lo hacen, para qué sirven, su equivalente en Basic normal, etc. Empezaremos por el más equívoco de ellos, el FORMAT8000, pues su nombre puede inducir a error a quien se aventure a adivinar su verdadera función.

FORMAT8000

Este programa cambia el formato de los programas grabados por el 700 a formato cargable por la serie 8000 ó 4000 (pero no los "traduce").

El 700 es capaz de cargar programas grabados por CPUs de la serie 4000 ó 8000, pero no a la inversa, es decir, el 8032 no puede cargar programas grabados por el 700. ¿Por qué? Cuando se ejecuta el comando SAVE o DSAVE, los dos primeros bytes que se graban en el disco son la dirección de memoria en donde el programa reside. Los comandos LOAD o DLOAD de las series 4000 y 8000 cargan los programas en la dirección indicada por estos dos primeros bytes, mientras que la serie 700 los ignora y carga los programas siempre a partir del byte 3 del banco 1 (para cargar programas en otra dirección dispone del comando BLOAD). Las direcciones de inicio son diferentes entre estas CPUs (3 para el 700, y 1025 para el 4000 y 8000). Por ello, un 700 puede leer programas de 4000 y 8000 (puesto que no hace caso de la dirección inicial), mientras que los 4000 y 8000 intentan cargar los programas del 700 a partir del byte 3, lo cual produce efectos realmente colapsadores.

Además, cada instrucción Basic tiene dos bytes apuntando a la dirección donde empieza la siguiente instrucción. Estos bytes, llamados links, se refieren a la posición de memoria absoluta, y no a la relativa respecto al inicio del programa. El formato de una instrucción Basic es:

```
[link] [numero] [texto instrucción 1...] [0]
[link] [número] [texto instrucción 2...] [0]
[link] [número] [texto instrucción 3...] [0]
[0,0]
```

El link y el número de línea ocupan dos bytes, mientras que la ocupación del texto Basic es variable según la longitud de la línea. Un byte a cero indica el fin de la instrucción; tres bytes consecutivos a cero indican el fin del programa. El link de una instrucción apunta al link de la instrucción siguiente hasta que éste link es cero.

En consecuencia, este utilitario debe cambiar tanto la dirección de inicio del programa como los links de encadenamiento entre instrucciones. El sistema de hacerlo es sencillo: leyendo el programa secuencialmente del disco (como si de un fichero se tratara) se efectúa una suma (o resta) a cada link para ajustarlo a la nueva dirección de inicio.

Una vez re-formatado, es posible que el nuevo programa no funcione correctamente en la serie 8000 si se usan comandos de MEC/DOS, posiciones de la página cero, funciones del editor de pantalla, o llamadas al sistema operativo (SYS). La traducción de todo ello debe hacerse manualmente.

En la línea 1020 del programa se asigna la dirección de inicio que tienen los programas en la serie 8000 (1025). Puede modificarse esta dirección para traducir, por ejemplo, programas de 700 a C-64; en este caso la dirección debe ser 2048. En el C-64, la re-formatación tiene otro handicap, que es el tamaño de la pantalla de ambos ordenadores: el 700 tiene 80 columnas mientras que el C-64 sólo tiene 40. Además debe utilizarse una unidad de discos compatible entre ambos equipos (por ejemplo los formatos de 4040 y 1541 son iguales), o utilizar el INTERPOD para cargar el programa en el C-64 desde un 8050, 8250, 9060 ó 9090.



UTILIDADES

Prints Sofisticados

Algunas Ideas para mejorar la presentación de los programas en Basic

Es frecuente que dado el esfuerzo que cuesta desarrollar un programa que funcione de forma segura no prestemos toda la atención necesaria al aspecto que tiene en la pantalla. Para solventar esto, en el artículo que sigue se dan algunas soluciones para hacer más variado el Comando Print.

No debe olvidarse que un programa "entra por los ojos" dado que el medio de comunicación con la persona que lo está usando es la pantalla. No está muy claro por qué algunos excelentes programadores suelen "pasar olímpicamente" del aspecto exterior de sus programas, quizá ésta es una costumbre que se arrastra desde los tiempos en los que los microordenadores sólo podían "sacar" por pantalla caracteres alfanuméricos y poca cosa más. En todo caso actualmente ya no existe excusa para que los programas sean pobres de presentación, incluso utilizando exclusivamente la instrucción PRINT como veremos a continuación.

En los ejemplos que vamos a dar se utilizan algunos trucos para mejorar la presentación en pantalla de una cadena de caracteres contenida en la variable A$. Se usan profusamente las instrucciones de BASIC, MID$, LEFT$ y RIGHT$. Quizá sea una buena idea resumir el funcionamiento de estas instrucciones.

LEFT$(A$,X) Aísla X caracteres de la cadena A$, contando desde la izquierda. De manera que si A$="ABCDEF" y X=3, LEFT$(A$,X)="ABC".

RIGHT$(A$,X) Realiza la misma función que LEFT$ pero contando desde la derecha. Así en el ejemplo anterior RIGTH$ (A$,X) en el caso de X=3 sería igual a "DEF".

MID$(A$,X,Y) Es algo más complicado: recoge Y caracteres empezando a partir del carácter X. Si asignamos A$="RSTUVWXYZ", X=5 e Y=4, MID$(A$,X,Y)="VWXY".

Antes de exponer los ejemplos recordamos que en éstos las funciones de manejo de cursor y gráficos se han sustituido por abreviaturas. Ver "CLAVE PARA INTERPRETAR LOS LISTADOS DE CLUB COMMODORE" (Pág. 6). El primer ejemplo es muy sencillo y sirve para darle un cierto "suspense" a la presentación:

```
100 A$="PRINT LENTO..."
110 FOR I=1TO LEN (A$)
120 FOR ZZ=1TO 200:
    NEXT ZZ
130 PRINT MID$ (A$,I,1);:
    NEXT I
140 PRINT:
```

En la línea 120 se controla la velocidad de la presentación con lo que comúnmente se denomina un bucle de retardo, valores menores de 200 aceleran la impresión y al revés.

Veamos el siguiente:

```
200 A$="PRINT DESLIZANTE"
210 A$="        "+A$
220 FOR I=1TO LEN(A$)
230 PRINT MID$(A$,LEN(A$)-I+1,1);
    "[CRSRU]"
240 FOR ZZ=1TO 25:
    NEXT ZZ: NEXT I
```

En este caso la presentación se desliza por la pantalla de una manera agradable, los espacios de la línea 210 determinan la posición final del mensaje en la línea, de manera que su número dependerá de la longitud del texto, del ancho de la pantalla y de la posición final que se desee.

PRINT A TROZOS es similar: las letras aparecen de una en una para acabar ocupando su sitio:

```
300 A$="PRINT A TROZOS"
310 FOR I=LEN(A$)TO 1STEP -1
320 FOR ZZ=1TO 24-LEN(A$)+1
330 PRINT TAB (ZZ)"    "MID$(A$,1,1);
    "[CRSRU]"
340 NEXT ZZ
350 NEXT I
```

Cambie el 24 en la línea 320 por un número menor (P. Ej. 10) si trabaja con un VIC-20.

El siguiente es un ejemplo de presentación de un gran efecto:

```
400 A$="UNA DEMOSTRACION DE
    SUPER PRINT"
410 A=LEN(A$):
    IF A/2<>INT(A/2) THEN A$="    "+A$:
    GOTO 410
420 FOR I=1TO A/2
430 PRINT TAB(20-I);LEFT$(A$,I);RIGHT$
    (A$,I)
440 PRINT "[CRSRU]";
450 FOR ZZ=1TO 50:
    NEXT ZZ
460 NEXT I
```

La línea 410 expande el mensaje a un número par de caracteres si es necesario. Aquí se usan simultáneamente las instrucciones RIGHT$ y LEFT$. Igual que antes en la línea 150 hay un retardo para controlar la velocidad de la presentación. Esta rutina trabaja mejor con cadenas de caracteres de longitudes superiores a 20. Si se trabaja con un VIC-20 se deberá cambiar el 20 de la línea 430 por un 10 y utilizar cadenas más cortas.

La siguiente rutina permite subrayar lentamente un mensaje en la pantalla:

```
500 A$="****DEMOSTRACION DE SUBRA-
    YADO LENTO*****"
510 L=40:
    IF LEN(A$)<40 THEN L=LEN(A$)
520 PRINT LEFT$(A$,L)
530 IF L=40 THEN PRINT "[CRSRU]";
540 FOR I=1TO L
550 PRINT CHR$(163);
560 FOR ZZ=1TO 30:
    NEXT ZZ
570 NEXT I
580 A$=RIGHT$(A$,LEN(A$)-L)
590 IF A$<>"" THEN 510
```

En este programa A$ se divide en longitudes de 40 caracteres (el ancho de la pantalla en el C-64) en las líneas 510, 520 y 580. Estas subcadenas se presentan con una línea en blanco entre ellas, que será donde se realiza el subrayado (con el CHR$(163) de la línea 550). El proceso continúa mientras exista contenido en A$ (línea 590). Cambie el 40 en las líneas 510 y 530 por 22 si trabaja en un VIC-20.

PRINT ALFABETICO es más complicado de explicar que de escribir. Para cada letra del mensaje se recorre al alfabeto hasta llegar al carácter del que se trata, parándose entonces:

```
600 A$="PRINT ALFABETICO"
610 FOR I=1TO LEN(A$)
620 M=ASC(MID$(A$,I,1))
630 IF M<65ORM>90 THEN PRINT
    CHR$(M);: NEXT I
640 FOR J=65TO M
650 FOR ZZ=1TO 20:
    NEXT ZZ
660 PRINT CHR$(J);"[CRSRL]";
670 NEXT J
680 PRINT "[CRSRR]";
690 NEXT I
```

En el código ASCII la letra A es 65 y la Z es 90, la línea 630 "filtra" los caracteres no alfabéticos y los imprime directamente.

PRINT RARO alterna los caracteres en video invertido de manera aleatoria:

```
700 A$="PRINT RARO"
710 PRINT A$
720 ZL=LEN(A$):
    DIM ZS(ZL)
730 FOR ZX=1TO 150:
    FOR ZZ=1TO 10:
    NEXT ZZ
740 ZR=INT(ZL*RND(1)+1)
750 PRINT "[CRSRU]";TAB(ZR-1);
760 IF ZS(ZR)=0 THEN ZS(ZR)=1:
    PRINT "[RVSON]";MID$(A$,ZR,1):
    GOTO 780
770 IF ZS(ZR)=1 THEN ZS(ZR)=0:
    PRINT MID$(A$,ZR,1):
    GOTO 780
780 NEXT ZX
```

Una matriz ZS() se utiliza para almacenar el estado normal/invertido de cada carácter (0 ó 1). ZR en la línea 740 "escoge" el carácter a invertir.

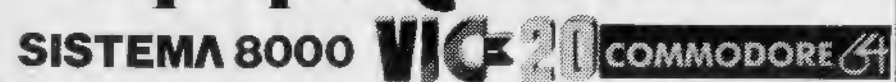
El último ejemplo hace que las letras suban una línea aleatoriamente. La técnica es muy similar a la utilizada en el caso anterior.

```
800 A$="PRINT AUN MAS RARO"
810 PRINT A$
820 ZL=LEN(A$):
    DIM ZK(ZL)
830 ZR=INT(ZL*RND(1)+1);
840 PRINT"[CRSRU]";TAB(ZR-1);
850 IF ZK(ZR)=0 THEN ZK(ZR)=1:
    PRINT "[CRSRU]";MID$(A$,ZR,1);
    "[CRSRD] [CRSRL]";" ":
    GOTO 830
860 IF ZK(ZR)=1 THEN ZK(ZR)=0:
    PRINT MID$(A$,ZR,1);"[CRSRU]
    [CRSRL]";" ";"[CRSRD]":
    GOTO 830
```

Nótese que hay un espacio entre las comillas de las líneas 850 y 860.

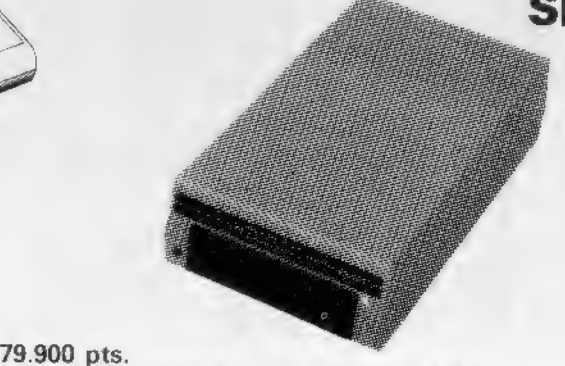
Una consideración final: No necesariamente el uso de técnicas de presentación poco trilladas hace que un programa sea bueno, el aspecto en pantalla de un programa es su carrocería, por muy vistosa que sea no iremos muy lejos si el motor no anda.

B.M.



Programas standard y "a medida" para equipos commodore

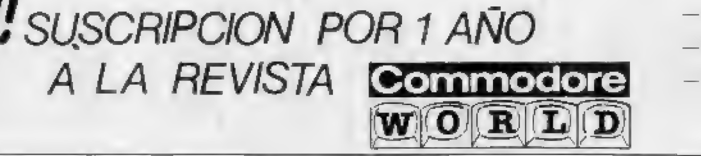
SISTEMA 8000 VIC-20 COMMODORE 64



— CONTABILIDAD (10MB)	— CONTABILIDAD	— QUINIELAS
— GESTION COMER.	— GESTION COMERC.	— AGENDA
— 9000 ARTICULOS	— STOCK ALMACENES	— COTIZACION BOLSA
— GEST. INTEGRADA	— VIDEO CLUB	— ETIQUETAS
— ALMACEN	— ENTRAPUNT	— PELUQUERIAS
— NOMINAS		— ETC.
— DIRECCION		
— AUTOVENTA		
— CONTROL SOCIOS		
— PRODUCCION		
— FINCAS		
— IND. CARNICAS		
— EMP. LIMPIEZA		
— COOPERATIVAS		
— TALLERES		
— COMPONENTES		
— PIENSOS		
— COLEG. PROFES.		
— CADENAS MONTAJE		
— PEDIDOS		
— FACTURACION		
— C. ORD. TRABAJO		
— VIDEO CLUBS		
— COMPETICIONES		
— BASCULAS Y PESAJE		
— ETC.		

COMMODORE 64	79.900 pts.	
	95.000 pts.	UNIDAD DE DISCOS 1.541
IMPRESORA GP-100 VC	54.900 pts.	
	25.000 pts.	PRG. CONTABILIDAD
	25.000 pts.	PRG. GESTION EMPRESA
	5.000 pts.	PRG. ETIQUETAS
TOTAL	284.800 pts.	

¡POR SOLO! 244.000





Apellidos
Nombre
Dirección
Población
D.P. Teléfono

EQUIPO		IMPORTE
OFERTA C-64		244.400
FORMA DE PAGO	TOTAL	
TALON BANCARIO REGISTRADO		
CONTRA-REEMBOLSO		
TOTAL		

PEDIDOS A:



B.M.
BASIC MICRO-ORDENADORES S.A.

Avenida Cesar Augusto, 72
Telefonos 235682 y 226544
ZARAGOZA-3



GALERIA DE SOFT

NOVEDADES

Microelectrónica y Control ha sacado 3 nuevos programas para el C-4. Introducción al BASIC, parte I, del que hicimos la reseña el mes pasado EASY CALC RESULT (20.000 pts.) y MACRO ASSEMBLER (7.500 pts.).

MACRO ASSEMBLER

Su referencia es ASM 6440 y se presenta en disco. Junto al compilador se incluyen una serie de programas complementarios, para facilitar el trabajo de edición, documentación y depuración de programas en código/máquinas. El contenido del disco es el siguiente:

ASSEMBLER 64: Es el compilador propiamente dicho.

CROSSREF 64: Este programa permite elaborar una referencia cruzada al final de los listados generados por el compilador.

DOS WEDGE4: Este es un utilitario que facilita el trabajo en disco.

DOS 5.1: En este fichero está contenido el código/máquina perteneciente al programa DOS WEDGE 64.

EDITOR 64: Amplía las prestaciones del editor de pantalla del COMMODORE 64 para facilitar la edición de textos/fuente para el compilador.

LOLOADER64: Amplía las pretensiones del editor de pantalla de COMMODORE 64 hasta facilitar la edición de textos/fuente para el compilador.

LOLOADER64 y HILOADER4: Estos dos programas se utilizan para cargar en memoria los ficheros creados por el compilador como código objeto.

MONITOR$8000 y MONITOR$C000: Los dos monitores son idénticos excepto en su lugar de carga. Permiten depurar los programas en código máquina.

COMANDOS DEL EDITOR: AUTO - CHANGE - CPUT - DELETE - FIND - FORMAT - GET - KILL - LIST - NUMBER - PUT.

COMANDOS DEL ASSEMBLER (ASSEMBLER DIRECTIVES): BYTE - WORD - .DBYTE - * - = - .PAGE - .SKIP - .OPT - .END - .LIB - .FILE - .MAC - .MND.

COMANDOS DEL MONITOR: ASSEMBLE - COMPARE - DISASSEMBLE - FILL - GO - HUNT - INTERROGATE - LOAD - MEMORY DISPLAY - NEW LOCATOR - REGISTER DISPLAY - SAVE - TRANSFER.

Como puede observarse, este paquete permite el desarrollo completo de programas en código/máquina para el COMMODORE 64, facilitando el uso de las características del equipo a máximo rendimiento.

Precio: 7.500 ptas.

EASY CALC RESULT

EASY CALC RESULT permite utilizar la pantalla del COMMODORE 64 como si fuera una gran hoja de papel donde se escriben cantidades y se efectúan operaciones entre ellas. Esta hoja está dividida en filas y columnas (63×254), de manera que cada cruce constituye una celda donde se escribe una cantidad, un texto corto para facilitar el uso, o una operación a efectuar entre dos o más celdas. En el momento en que se ha terminado de escribir una cantidad en una celda se efectúan todas las operaciones de la hoja, así es muy fácil preparar presupuestos donde sea necesario determinar rápidamente "QUE PASARIA SI..." y ver inmediatamente el resultado de alterar uno o varios factores en un cálculo completo. En cualquier momento se puede copiar el contenido d la hoja de trabajo en la impresora y se pueden preparar gráficos de barras (Histogramas) que, a su vez, se pueden volcar en impresora. La HOJA DE TRABAJO y su contenido se pueden almacenar en disco o cinta para trabajos posteriores.

Este programa es una versión simplificada del que actualmente se comercializa bajo la denominación CALC RESULT (ADVANCED). El cartucho se suministra con un completo manual en castellano.

Precio: 20.000 ptas.

PROGRAMAS VIC-20
Programas en Disco

Pesetas

Agenda.. 5.000
Permite contener 4 páginas de información de 114 personas, entidades, etc. Pudiendo añadir, cambiar o borrar la información que se desee. En castellano, necesita ampliación de 8K. Presentado en caja.

QSL.. 3.000
Registro de contactos para radioaficionados, versión en disco. Necesita ampliación de 8K. Presentado en caja.

Test Demo.. 3.000
Programa de test para la unidad de disco. Nota: este disco es el que se suministra con la unidad de disco VIC-1540.



Assembler 5.000
Editor y compilador en BASIC para rutinas en lenguaje máquina del 6502. Necesita ampliación de 3K. Acompañado de manual en inglés.

English languaje 2.500
Versión en disco del programa en cinta, dado que al manejar ficheros resulta mucho más rápida su utilización. Necesita ampliación de 8K. Presentado en Caja.

Quiz Master 2.500
Versión en disco del programa en cinta. Necesita ampliación de 8K. Presentado en caja.

Matemáticas 1 (nivel BUP) 2.500
Versión en disco del programa en cinta. Necesita ampliación de 8K. Presentado en caja.

Simplicalc 13.000
Hoja de Trabajo Electrónica 60 columnas por 100 filas Max. Necesita ampliación de 16 K.

Vic File 13.000
Base de datos para el VIC 20 con posibilidad de definir el formato de los campos. Necesita ampliación de 16K. Ideal para fichero de clientes.

Vic Writer 13.000
Proceso de texto muy potente y rápido. Necesita ampliación de 8 ó 16K.

Continúan en el próx. nº

SOFTWARE COMMODORE 64
Utilitarios y Lenguajes

Pesetas

Monitor lenguaje máquina 10.000
Este cartucho permite entrar y depurar rutinas en código máquina. Dispone de un conjunto de instrucciones muy potente. Ideal para iniciarse en las técnicas de programación en lenguaje máquina. Presentado en cartucho.

Forth 20.000
Nuevo lenguaje de alto nivel y gran potencia que permite definir las propias instrucciones. En una aplicación se puede trabajar en rutinas críticas con el Assembler que contiene. Presentado en cartucho.

Logo 16.000
Es un lenguaje que utiliza Procedimientos (Procedures) que pueden ser definidos por el usuario a partir de los existentes en el lenguaje (Primitives), por lo que resulta de gran sencillez y posee una gran potencia. Ideal para enseñanza. Alta resolución y Sprites incorporados. Presentado en disco.

Pilot 16.000
Lenguaje específico para educadores. Permite definir el juego de caracteres y manejar Sprites, alta resolución y sonido fácilmente. Presentado en disco.

Macro Assembler 7.500
Potente paquete editor ensamblador que le permitirá desarrollar y depurar fácilmente sus programas en código máquina. Presentado en disco

Programmer's Utilities 7.500
Paquete de utilidades. Contiene editor de caracteres, editor de sprites, monitor lenguaje máquina y otros programas que le podrán ayudar a programar más rápidamente. Presentado en disco.

Turtle Graphics II 15.000
Subconjunto del lenguaje LOGO. Es ideal para principiantes en edades comprendidas entre 6 y 60 años. Presentación en cartucho, utiliza indistintamente disco y cassette.

Master 25.000
Base de datos de gran potencia. Incorpora nuevas instrucciones que facilitan el trabajo en cuatro grandes áreas:
—Edición y formatación de datos en pantalla.
—Edición y formatación de datos en impresora.
—Manejo de ficheros
—Incorpora instrucciones de BASIC 4
Facilita el desarrollo de aplicaciones profesionales. Presentado en disco. Manual en castellano.



Magia

Hemos contactado con los espíritus más poderosos del mundo de la informática para que nos trajeran esta selección de consejos útiles, indicaciones y métodos. Estas maravillas de la magia les van a encantar con unos métodos fantásticos para que Vd. aproveche al máximo su sistema.

La MAGIA son trucos, la MAGIA es divertida.
La MAGIA es hacer lo que nadie se ha atrevido.

La MAGIA es una columna mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones.

Cada mes, MAGIA les trae trucos breves y útiles de informática procedentes de todo el mundo - trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés. Buscamos material nuevo o renovado que resulta ser de valor actual para usuarios de equipos Commodore y que puede utilizarse con un mínimo de tiempo, esfuerzo o conocimientos teóricos.

MAGIA resulta ser la fuente más completa de información para la informática práctica, además de ser un foro internacional para compatir trucos con otros aficionados. Envíe sus trucos a:

COMMODORE WORLD
Pedro Muguruza, 4
Madrid-16

La revista "Commodore World" sorteará seis paquetes de software en julio y diciembre entre todas las contribuciones publicadas.

A continuación escribo alguna observación que he hecho con mi "COMMODORE-64" y que se puede incluir dentro de la sección de "MAGIA":

Si pongo lo que son las instrucciones en minúsculas y lo que son las variables en mayúsculas, al "correr" el programa me aparece un error en las instrucciones en las que he puesto las variables en mayúsculas. El error se debe a que esas variables no aparecen en el listado.

De donde se deduce que al colocar en un mismo programa minúsculas y mayúsculas, estas últimas deberán ir siempre entre comillas, de lo contrario se generará ERROR.

Esto conviene recordarlo cuando se trabaja con mayúsculas y minúsculas en un mismo programa.

También deben estar entre comillas todos los caracteres gráficos de la parte frontal de las teclas.

J. M. V.
(Iniciales sólo por deseo del colaborador).



Amigos de **Commodore World**, os envío la siguiente colaboración para el apartado de "Magia".

SUBRUTINA MAGICA DE FACTORIALES.

Después de observar en el manual del VIC las maravillosas posibilidades que tiene esta máquina para crear funciones trigonométricas complejas, me quedé un poco decepcionado al ver que no se nombraba para nada funciones tan importantes como el Factorial (!), fundamental en combinatoria (donde se pueden hacer programas muy interesantes).

La fórmula a utilizar es la siguiente:

```
5000 INPUT A
5010 C=INT(A-1):B=INT(A)
5015 IF A<0 OR A>33 THEN 5100
5020 FOR J=CTO1 STEP-1
5030 B=B*J
5040 NEXT J
5050 IF A=0 OR A=1 THEN B=1
5060 PRINT B
5070 RETURN
5100 PRINT "LO SIENTO. ESTE FACTORIAL NO PUEDO
     CALCULARLO".
5110 RETURN.
```

Desde luego no voy a revelaros la"pócima secreta, pero me parece que está claro que la base del conjuro es el bucle del paso 5020.

David Pérez Corral - 15 años.
Passeig de Sant Joan 171, 5º 4.ª
Barcelona-37

NOTA: Si éste truco no os parece interesante o puede ser mejorado, por favor, comunicádmelo.

○ ◯ ○

Amigos de **Commodore World**:

En primer lugar, os felicito por la revista que estáis realizando, la cual parece interesante para la multitud de usuarios del Commodore. Y aprovecho la ocasión para enviaros un pequeño **truco** sobre VIC-20: La forma normal de detener un programa a la espera de que el usuario pulse una tecla es mediante la sentencia GET: 100 GET A$: IF A$="" THEN 100.

Esta sentencia ocupa 21 bytes, además de los que ocupe la cadena A$. Una forma elegante de hacer lo mismo economizando algunos bytes es mediante WAIT:

100 POKE 198, 0: WAIT 198,1. En caso de que interese saber qué tecla ha sido pulsada, puede usarse GET A$.

Felicitándoos una vez más por el buen trabajo que seguro vais a realizar, y ofreciendo toda la ayuda que me sea posible, me despido.

Hasta pronto.

Alfredo Sindin Valero.
Avda. Logroño, 42 - Villa Esther.
Madrid-22.



(VIC-20) - Si apretamos la tecla RUN/STOP y la tecla SHIFT/LOCK simultáneamente (a la vez) y suavemente aparece la palabra LOAD. Pero cuidado no dejar la tecla SHIFT/LOCK apretada; en este caso repetir de nuevo ya que no siempre da resultado.

Entonces apretamos normalmente la tecla PLAY del Datacassette y ahí **la novedad reside en que una vez cargado el primer programa encontrado lo ejecuta automáticamente.** Es decir, sin necesidad de teclear la palabra RUN. Es bastante cómodo.

¡Saludos! de

Cristobal Mora Pol
C/Reina María Cristina, 54-1º 3
Palma de Mallorca (Baleares)

Para posicionar el cursor en un sitio determinado, existe otro truco además de los publicados en el número anterior, es el siguiente:

Para el VIC-20: PRINT "HOME" TAB (22 × FILA + COLUMNA); "funciona".

para el C-64 debéis sustituir el 22 por 40.

Marisa L. Alvaro
Portolagos, 7
Alcorcón.



Queridos amigos: Continúo mi felicitación para este 2º número de la revista que no tiene desperdicio, creo que los "Commodorianos" estamos de enhorabuena. Os envío un saludo y unos Pokes para "Magia".

En la posición 774 tenemos un Poke muy interesante, relacionado con al forma de Listar del Commodore 64. Haciendo Poke 774,255 impide la representación del Listado; Poke 774,0 nos Lista únicamente los números de sentencia sin posibilidad de ver el contenido de cada línea. Poke 774,34 llena la pantalla con un Scroll de extraños signos y finaliza con CLR; Poke 774,207 le dará continuos errores de línea. Ensayando con distintos valores se descubren cosas realmente curiosas e introduciéndolas en un programa que se autoejecuta impide su listado a los curiosos que no conozcan la palabra "Mágica" que vuelve todo a la normalidad; Poke 774,26.

Domingo Márquez Gallego.
Parque Vosa, 3. 4º C.
Móstoles (Madrid)
Teléf.: 614 21 48



Os mando el siguiente programa para la sección Magia.

Consiste en crear un juego de caracteres de doble altura y formatear pantalla para esta presentación, teniendo en cuenta que no caben todas las columnas en la pantala, sólo 13.

Esta versión es para el VIC-20 que aunque "chupa" mucha memoria funciona en un VIC-20 sin expansión y se puede utilizar en un programa de presentación que cargue automáticamente al siguiente.

Siento mucho no ofrecer una versión para el C-64 pero no tengo su mapa de memoria.

```
5 P1=PEEK (657): P2=PEEK (56): P3=PEEK (55): P4=PEEK
  (36869): P5=PEEK (36867): P6=PEEK (36865)
6 POKE 657,128
7 POKE 56,20: POKE 55,0
8 A=5120: B=7678
9 FOR T=A TO B STEP 2
10 P=PEEK (32768+(T-A)/2)
11 POKE T,P: POKE T+1,P
12 NEXT T
13 POKE 36869,253
14 PRINT CHR$ (147): POKE 36867, 29: POKE 36865,29
15 PRINT "JORGE RODRIGUEZ"
16 STOP
17 POKE 657, P1: POKE 56,P2: POKE 55,P3
18 POKE 36869, P4
19 POKE 36867, P5: POKE 36865, P6
20 PRINT "(CLR) (4 espacios) "CHR$(34)"Nombre del pro-
   grama" CHR$ (34)
21 POKE 631,19: POKE 632,131: POKE 198,2: NEW
```

El programa en sí está compuesto por las líneas 6 a la 14.

Las líneas 17, 18, 19 reestablecen al VIC a la normalidad con los valores cargados en la línea 5.

La línea 20 imprime en pantalla el nombre del programa a cargar desde cinta, y la línea 21 ejecuta el LOAD del programa.

Para leer un programa desde el disco modificar la línea 20 por 20 PRINT"(CLR) (4 espacios)"CHR$(34)"Nombre del programa"CHR$(34)",8"

Vuelvo a advertir: el programa de los caracteres a doble altura sólo está comprendido entre las líneas 6 a la 14, siendo la línea 15 una demostración. Este programa sólo funciona en el VIC-20.

Las líneas 5, 17, 18, 19 son para reestablecer la configuración inicial del VIC.

Las líneas 20 y 21 son para cargar desde un programa otro y que se ejecute automáticamente. Esta técnica funciona en el VIC-20 y en el C-64.

Bueno después de esto me despido esperando no causar muchas molestias y que me contestéis lo antes posible.

Saludos.

Jorge Rodríguez Piñeyro.
Polígono San Pablo. Barrio C.
Torre 2
Sevilla-7

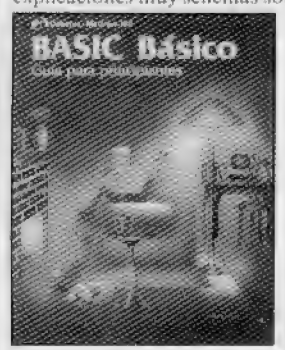


BASIC Básico

Ideal para cualquier edad sin necesidad de tener conocimientos previos sobre informática.

En el prólogo y primer capítulo, a modo de cuento, nos presenta a Gerardo que es una persona totalmente ajena a la informática y decide introducirse en este mundo. La toma de contacto la hace con un pequeño retroceso histórico para ver el nacimiento de los primeros ordenadores y las aplicaciones para las que fueron diseñados, así como sus prestaciones.

En el segundo capitulo, ya en tono serio, nos da una serie de definiciones fundamentales con explicaciones muy sencillas sobre su significado, y nos presenta el equipo físico (hardware) y los programas (hardware), cómo se almacenan y cómo se imprimen. Nos presenta la gama de periféricos que se pueden utilizar con un ordenador.



El tercer capitulo es el primero que trata de BASIC propiamente dicho y nos presenta las primeras instrucciones que debemos conocer (PRINT, LIST, RUN y NEW). En el cuarto conocemos las variables como medio de almacenamiento de datos. El 5º demuestra el modo de introducir datos (INPUT), y en sucesivos capitulos trata las sentencias condicionales (IF/THEN), los bucles (FOR/NEXT), números aleatorios, sentencias READ/DATA, subrutinas (GOSUB), etc., terminando con un capítulo que trata sobre el futuro de las computadoras.

Todos los capitulos van muy detallados y con muchos ejemplos de todo, y todo lo que aparece en este libro es válido para cualquier marca de microordenador que trabaje en BASIC. Al final de cada capitulo existe un cuestionario para comprobar que se ha entendido todo lo tratado en él.

Basic Básico
Hume Fox - David Fox. 278 Páginas
Editorial McGraw Hill - España
Sta. Beatriz, 4, Madrid-18
Teléfono: 01-251 32 03

VIC-20
Guía del Usuario

Libro útil para aprendizaje y consulta que explica detalladamente todo lo que el manual del usuario deja sin aclarar.

Enseña todo lo referente al VIC-20, desde cómo conectarlo hasta cómo utilizar al máximo todos sus periféricos.

Comienza explicando cómo utilizar el VIC-20 en el modo de comandos directos y en el de programa, cómo editar en pantalla, cómo utilizar el teclado correctamente, etc.

En el capítulo de programación describe todos los comandos y las funciones del BASIC del VIC-20 con detalle.

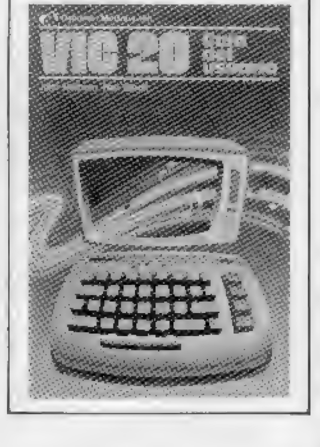
En "Programación avanzada" se estudian las cadenas numéricas y alfanuméricas, así como el formateado de entradas y salidas, junto con subrutinas de manejo de datos, el reloj de tiempo real y los números aleatorios.

El 5º capítulo analiza el funcionamiento de los mandos "joystick" y la utilización del teclado para controlar los juegos.

El controlador de vídeo, los juegos de caracteres y los mapas de memoria de pantalla y colores son el objeto de estudio del 6º capitulo además de animación de juegos, caracteres personalizados en alta resolución, programas con caracteres definidos por el usuario, gráficas de alta resolución, multicolor, etc.

El 7º es un estudio detenido sobre el sonido y su generación en el VIC-20.

El capitulo 8º nos presenta la gama de periféricos del VIC-20,



deteniéndose en el estudio de los comandos de control del cassete, disco, modem o impresora.

En los apéndices aparecen un diagrama de bloques del VIC-20, el mapa de memoria del VIC y la tabla de las rutinas KERNAL, los conectores de entrada-salida, tablas de conversión, códigos de caracteres y sonido, mensajes de error, sentencias y funciones BASIC.

En resumen se trata de un libro muy útil para todo usuario del VIC-20, que quiera tener más información sobre el funcionamiento y posibilidades de su aparato y prefiera una alternativa a la guía de referencia del programador, ya que sus contenidos son semejantes aunque varia la presentación y exposición de las diferentes materias.

Vic-20, Guía del Usuario
John Heilborn - Rantalbott
400 páginas
Editorial McGraw Hill España, S.A.
Sta. Beatriz, 4 - Madrid-18
Teléfono 01-251 52 03

Guía del Ordenador Personal PET/CBM

Describe los siguientes ordenadores de Commodore:

PET 2001/8K, 2001/8N, 2001/16N, 2001/32N, CBM2001, 16B, 2001/32B y las series CBM 4000 y CBM 8000, sus teclados, pantallas, periféricos y conexiones.

Las versiones BASIC estudiadas son las 1.X, 3.X, y 4.X, donde X es el número de la subversión.

Contenidos:

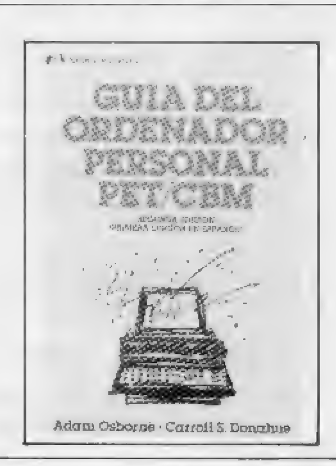
Introducción a los ordenadores CBM, funcionamiento, edición en pantalla, programación, caracteristicas especiales, dispositivos periféricos, información del sistema, BASIC CBM y apéndices.

Entre los apéndices figuran los códigos de caracteres, mensajes de error, bibliografía sobre BASIC y publicaciones relacionadas con CBM, Tablas de conversión, variaciones del nivel de revisión 2 de memorias ROM, mapas de memoria y tablas sobre las direcciones de rutinas BASIC incluidas en la ROM.

Quien ya conozca el estilo de Adam Osborne sabe que todos los capitulos están tratados con mucho detalle, ilustrados con cuadros y tablas, e incluyen ejemplos prácticos de cada explicación teórica.

Es por tanto un buen libro de consulta para los usuarios de los PET/CBM y útil para todo el que desee conocer algo más sobre el BASIC de estos.

Guía del Ordenador Personal Pet/Cbm
Adam Osborne y Carroll S. Donahue
530 páginas
Editorial McGraw-Hill de España, S.A.
Sta. Beatriz, 4 - Madrid 18
Teléfono: 01-251 52 03



CARTA BLANCA



AYUDA!!

Somos una sociedad de tipo cultural y acabamos de formar un club de usuarios de microordenadores, enfocado principalmente al Commodore y al Sinclair, por ser los más vendidos en España a nivel personal.

Desearíamos saber en qué medida pueden ustedes colaborar con nosotros, pues partimos de cero, y sobre todo en la coordinación de cursos basic ya que disponemos de amplios locales y personal cualificado para impartir clase de programación.

También agradeceríamos que en el próximo número de la revista dedicasen unas líneas a la información sobre la creación de dicho club.

Por nuestra parte, no dudaremos en enviarles los programas que aquí vayan surgiendo, así como trucos de programación que aquí se descubran, y las actividades que se estén llevando a cabo en cada momento.

Esperando su grata contestación, reciba mi más cordial saludo.

LUIS MARCONI SUAREZ VIEITES
SECRETARIO
DEL CIRCULO OURENSAN-VIGUES.

En primer lugar, quisiera felicitarle por la nueva revista y pasar a expresarle el motivo de nuestra carta.

Nos hemos juntado unos usuarios de Commodore, aquí en Albacete, pues creemos que sería muy positivo el crear un club en esta ciudad, que sería muy beneficioso para todos, y es por esto que les pedimos por favor, que nos ayuden en lo que se refiere a información sobre funcionamiento, sonido y lenguaje a máquina.

También quisiéramos que publicaseis algo sobre nosotros, informando que necesitamos socios para dar vida a este club de Albacete, ya que no conocemos a la mayoría de los usuarios que pueda haber aquí.

También nos gustaría algo de información en cuanto a programas de música para el C-64.

CLUB DE COMMODORE DE ALBACETE
Fernando Martínez Guerrero
La Roda, 39-5º D
ALBACETE

Nosotros somos la revista, no la casa Commodore, por tanto sólo podemos ayudaros con la publicación de todo aquello que los distribuidores de Commodore en España nos envíen. Cualquier información que Microelectrónica y Control o cualquier otra casa nos remita la publicaremos para que pueda ayudaros, así como las notas, trucos, etcétera, que otros lectores nos manden.

Nosotros no podemos enviaros manuales ni libros sobre las rutinas internas de los equipos ya que no disponemos de ellos para venta al público.

Podéis pedir información más detallada sobre equipos, soft y publicaciones, a los distribuidores de Commodore y librerías especializadas para que ellos puedan facilitaros algo más o deciros dónde podéis conseguirlo. Podéis comenzar por contactar a los de Albacete y Vigo y a los que aparecen en la revista.

En cuanto a la ayuda que podemos prestaros en los cursos que habéis organizado, podéis preguntarnos todas las dudas que puedan surgir, nosotros trataremos de aclararlas y las publicaremos para que sirvan de guía a los demás clubs y usuarios independientes.

Estamos deseando recibir vuestras colaboraciones.

GRAFICOS PARA EL 64

Vengo buscando desde hace bastante tiempo sin conseguirlo, la dirección donde pueda adquirir un cartucho (de los que se introducen directamente en el ordenador, que cumpla al menos como el Superexpander lo hace con el VIC-20; naturalmente para el Commodore 64).

CARTA DEL MES

Soy suscriptor de los "viejos" desde el número uno. Soy suscriptor de COMMODORE INTERNACIONAL (Inglesa) y de un montón más de revistas de informática.

Quisiera que ustedes fueran distintos de las demás revistas de informática, que todas terminan siendo del Vic-20 ó del ZX como si no existiesen más ordenadores.

QUISIERA QUE HICIERAN HONOR AL NOMBRE QUE LE HAN PUESTO A LA CRIATURA Y NO SE OLVIDEN QUE COMMODORE SON TAMBIEN EL COMMODORE 2000 y EL 3000 ? TAMBIEN EL 4000, el 8000 y todos los demás y que como DIVERTIMENTO EXISTEN EL VIC 20 y EL COMMODORE 64.

Tengo un COMMODORE 3032 y es más difícil conseguir software y aplicaciones como extensiones del port para comandar y controlar procesos, etcétera, etc., que un ciego ver.

He contactado con el representante del Commodore en esta provincia, que es un buen amigo y me informan que MICROELECTRONICA NO QUIERE SABER NADA DE LOS 2000 y 3000.

Dejo colgada mi pregunta, ¿COMMODORE WORLD TAMBIEN SE VA A OLVIDAR DE LOS ESPAÑOLES QUE CON GRANDES SACRIFICIOS Y SIENDO LOS PIONEROS, NOS EMBARCAMOS EN COMPRAR UNOS EQUIPOS CAROS Y HOY NOS VEMOS DESCOLGADOS DEL CARRO DE LA INFORMATICA?

LES SEGUIRE POR UN PERIODO DE UN AÑO. Si ustedes son conscientes del nombre que le han puesto a su revista, sin duda, tendrán en mí un suscriptor por mucho tiempo.

ARTEMIO GONZALEZ PEREZ (ARGOGAS)
FERNANDO AROZENA QUINTERO, 1
(POLIGONO COSTA SUR)
TELEFONOS 22 42 44-45
SANTA CRUZ DE TENERIFE

Aunque no ayude gran cosa, podemos decir que estamos de acuerdo, y que la situación es indiabladamente frustrante. La situación del pionero es siempre altamente injusta. No pretendemos disculpar a la casa Commodore al decir que la situación de los PETS es la misma que la de todos los equipos pioneros de cualquier otra firma de ordenadores, pero la triste realidad es esa. El micro (de cualquier marca) nació antes de que las firmas creadoras fueran conscientes del futuro impacto y necesidad de un amplio soft.

El impacto se produjo, la informática se desbordó y los equipos se sucedieron unos a otros a una velocidad no anticipada. Las casas de soft nacieron... para el último equipo, olvidando a los pioneros.

En Microelectrónica, con quien hemos hablado, reconocen este problema pero se reconocen impotentes, sin embargo, nos dicen que su Departamento de Soft atenderá todas las consultas que se le hagan (Sres. Masats o Navarro) e intentarán ayudar en lo que puedan.

Nosotros por nuestra parte vamos a hacer lo imposible para sacar información.

Desde aquí hacemos una llamada de S.O.S. URGENTE a todos los distribuidores nacionales o extranjeros y a todos nuestros lectores para que nos envíen, tanto a nosotros como a tí directamente, toda información por pequeña que sea sobre los PETS.

Esperamos y deseamos poder darte el servicio que necesitas.



...Y... SEAMOS

que me permita el uso de las 4 teclas programables, y, una cómoda programación para el trazado gráfico de funciones matemáticas. Hasta ahora los métodos empleados en alta resolución son incómodos y largos, les quedaría muy agradecido si pudiesen ayudarme a encontrarlo.

ALBERTO VALDENEBRO MORAÑO
CALLE REAL, 82-7º B - TELEF. 50 86 37
(EDIFICIO CARTHAGO NOVA)
CARTAGENA (MURCIA)

Próximamente Commodore pondrá a la venta un cartucho del estilo del Super-expander pero para el C-64. Actualmente ya tiene disponible en disco el "Turtle graphics II", el "LOGO" y el PILOT, que son tres discos que permiten trabajar al ordenador en lenguajes específicos para gráficos.

En este mismo número tienes un programa que permite asignarle funciones a las teclas de funciones.

La información sobre cartuchos y programas disponibles en el mercado irá apareciendo poco a poco en la revista en la sección "Galería del Soft".

AMPLIACION DE RAM

Aprovecho para felicitarles por la presentación y contenido de su nueva revista. Tengo el siguiente problema: Poseo un VIC 20, con la máxima ampliación de memoria. Pero al moverse la memoria de pantalla, al conectar los cartuchos de 8 y 16K RAM, queda un área de unos 3K que se desaprovecha como memoria accesible en Basic. Les agradecería me indicaran, si es posible, mediante alguna instrucción rescatar este área de memoria, que me sería muy útil, por dedicar mi equipo fundamentalmente a cálculos técnicos de ingeniería civil y edificación.

Saludos y gracias por todo.

LUIS LUCERO AOIZ
C/ ALEJANDRO GONZALEZ, 5-1º B
MADRID-28

En el número 2 ya contestamos a una pregunta similar, repetimos para los que no tengan el número 2.

Cuando se conectan las ampliaciones de 8 ó 16 KRAM al VIC-20, si tienes conectado el Super-Expander no sólo no tienes los tres KRAM disponibles en BASIC; sino que te quita algunos bytes de la memoria que tendrías si lo quitases.

Estos bytes son los ocupados por los "strings" que se le asignan a las teclas de función, y los quita del final de la memoria.

Los 3 KRAM del super-Expander siguen estando disponibles, pero no los puedes usar para programas BASIC, sólo para programas de lenguaje máquina. No sé de ningún POKE misterioso que te los pueda devolver. Si alguien lo sabe que nos lo envíe y lo publicaremos.

PROBLEMAS CON LA CLAVE

Muy señores míos: En el número 1 de su revista listan ustedes el programa "Tiro al blanco" y aparecen una serie de letras entre corchetes, como (CRSR DN) y (3CRSR DNS).

El grupo ERSR entiendo perfectamente su significado, lo que les agradecería enormemente es la explicación de los grupos (DNS) y (DN).

Dándoles las gracias por adelantado y en espera de sus gratas noticias. Se despide de ustedes.

FDO. JOSE V. BAYO GARCIA
ISIDORO DE LA CIERVA, 3
TELEFONO (968) 21 45 40
SUC. JERONIMO DE RODA, 1
TELEFONO 24 33 93
MURCIA

Por tratarse del primer número, utilizamos dos claves diferentes para los listados, en el segundo ya lo corregimos adaptándonos a la clave que todos conocíais del Club Commodore. El significado de (CRSR DN) es cursor abajo (cursor DownN), y el de (3CRSR DNS) es tres veces cursor abajo.

MAL POETA ...Y ALGO "PIRAO" TAMBIEN

Amigos de la Commodore: (que bien me ha quedado ¿eh!)

Comenzaré diciendo que soy nuevo en éste club, pues no hace mucho tiempo que adquirí mi propio VIC-20.

Os daré algunas de mis opiniones sobre este primer número de "Commodores World" (ya que he tenido la "chorra" de suscribirme en el momento justo, para recibir este número 1):

1.ª opinión: creo que deberíais tener en cuenta los que se incorporan a este fascinante mundo de la informática (¡olé! yo para poeta) los cuales están algo "verdes" con lo referente al sistema (todo hay que decirlo, yo soy uno de ellos), por lo que os agradecería unas explicaciones lo más claras posibles y a poder ser ejemplos de lo que resulte más lioso.

A título de ejemplo, os diré que me estuve rompiendo la cabeza con el articuo de este primer número titulado "Personalizar tu VIC con gráficos de juegos"; al final logré descifrarlo y fue gracias al ejemplo. Pero después vinieron las gráficas de alta resolución y ahí la palmé, pues no tenía ni idea de las formulitas que poníais.

Con todo este rollo os quiero pedir un ejemplo de programa que utilice las gráficas de alta resolución y caracteres multicolores.

2.ª opinión: ¡procurad evitar los errores de imprenta o del propio editor! ya que la "meva merced" no estoy muy enterado de la materia para subsanar los errores en un programa que trabaje con POKEs algo rarillos. Si no conseguís este objetivo por lo menos crear un apartado de erratas.

Pregunta: ¿Dónde puedo encontrar información de memoria de 16K?

Demanda: ¿podéis publicar un programa en lenguaje máquina y después su equivalencia en BASIC?

JAVIER SÁENZ PULIDO
C/ MÉJICO, 30-4º B-4

¡Vale!, en primer lugar quiero decirte que procuraremos poner ejemplos en todo lo que podamos. En segundo lugar desde el principio hemos intentado evitar las erratas, per el duende de imprenta es prácticamente imposible, siempre puede colarse un gazapo en cualquier sitio, pero intentaremos localizar todos los que se cuelen y ponerlos en el apartado "meteduras de pata" que como puedes ver aparece en este número por primera vez (¡ojalá pudiera ser la última!).

Respecto a tu pregunta siento decirte que la respuesta es no. Pero podrías conectarlo si le hicieses unos cortes en las pistas y unos puentes para cambiar la dirección en que están situados. No es sencillo si no tienes algo de experiencia con el soldador, y por supuesto anularía la garantía...

Sobre los programas en lenguaje máquina y basic equivalentes, lo intentaremos próximamente... pero será con programas sencillos para ficilitar la comprensión de ambos.

Intentaremos publicar en el futuro (¡no nos pidas fecha!) algún programa que utilice gráficos de alta resolución. Mientras tanto, lo dejamos aquí como un desafío a los lectores y clubs. A ver quien nos envía el primero.

Me ha gustado la página 42 "Más allá del manual", que supongo seguirá, pues para los que comenzamos todas las ayudas son pocas.

Parece que falta una sección con bibliografía, para saber qué libros se pueden comprar y cuáles no y un poco de su contenido.

Me ha sorprendido la contradicción entre una frase del MANUAL DEL USUARIO, que viene con el Commodore 64 y que dice: "Commodore se ha volcado para que sus manuales sean conocidos por su facilidad en la lectura, uso y comprensión.", y los errores que van apareciendo. Por ejemplo en la página 44 al explicar la línea 70. Creo que la flecha está mal puesta. En la página 61, arriba en el ejemplo de POKE creo que falta un número. En la página 106 anuncia una bibliografía con más de 20 libros, que nunca aparece. (P. 152 manual Inglés del 5 × 64.)

Todo esto puede parecer una referencia a cosas pequeñas, pero que a un principiante pueden desorientar. Quizá pueda quedar suplido con la revista.

No vendría mal en la revista una sección dedicada a un glosario, pues constantemente se leen u oyen palabras, que no se conoce su significado.

Un cordial saludo.

F. JAVIER LOPEZ IGLESIAS
MARIANO D. BERRUETA, 17-3º J
LEON

Gracias por enviarmos los errores que has encontrado en el manual. Creo que Microelectrónica y Control ha preparado una nueva edición corregida del manual, pero al publicar los errores que encontréis, los demás usuarios tendrán más liso el camino si tienen un manual antiguo.

Desde ahora vamos a dedicar una pequeña sección a los errores de los manuales, en ella iremos publicando todos los errores que encontremos entre todos (vosotros y nosotros)...

En la página 44 está mal la flecha, debe indicar a la línea 30.

En la página 61 debería decir POKE 53281,7.

La bibliografía a la que hace referencia en la página 106 imagino que no parece por ser toda en inglés. Tengo el manual original y en éste sí aparece. En español sólo tienes los manuales de los ordenadores y periféricos y las guías de referencia del programador del VIC-20 y C-64, así como los cursillos de BASIC de Commodore.

Además vamos a publicar reseñas de todos los libros que las editoriales nos comuniquen que existen, como ejemplo en este mismo número aparecen tres de McGraw-Hill España, el nueva sección bibliográfica. De tomo y lomo (pág. 48).

PREGUNTONES



El Glosario también comienza en este número, como ya habrás visto y, finalmente —Mas Allá del Manual—, sí será una sección fija, que publicaremos en todos los números.

FOTOGRAFIA

Me gustaría saber cómo se podría controlar una ampliadora fotográfica utilizando el "C-64". Una explicación entre el interface adecuado y (su conexión) la ampliadora que sea. Si es que es posible, es decir, que haya algún lector o socio que pudiera decírmelo.

J. M. V.

En el número uno de Commodore World apareció un artículo sobre cómo utilizar el port de usuario del VIC-20. El port de tu C-64 es prácticamente igual, pero varían las direcciones y algunas peculiaridades del circuito integrado utilizado (CIA). Si desea información detallada puedes encontrarla en la página 293 de la guia de referencia del programador. Si podemos en el futuro dedicaremos un artículo a este port, pero ahora mismo no puedo prometerte nada.

En el mercado nacional tienes un accesorio que te permitiría utilizarlo para controlar la ampliadora, se trata del "controlador C-8" de Elektrocomputer.

J. M. V.
(INICIALES SOLO POR DESEO PROPIO)

MEA CULPA... MEA CULPA

He mandado un programa, el cual ha sido publicado en su revista. Yo no la he recibido aún, pero me he enterado por un amigo que la recibe. En ella, está el programa, y una breve nota que dice "mándanos tu dirección". Creo que ya la tienen, soy uno de los muchos que están suscritos a su revista, pero de todos modos, se la pongo:

LUIS CARBALLO TABOADA
C/ CARDENAL QUEVEDO, 35-5º
ORENSE

El programa es Carotena publicado en el número 1 de Commodore World.

Lamentamos que hayas recibido el número 1 tan tarde. Al pasarnos Microelectrónica y Control su fichero de suscriptor de Club Commodore hubo un problema de etiquetaje que afectó a unos veinte suscriptores, entre ellos tú. Como habrás visto, ya está subsanado.

¡PRESUME CON TU AMIGO!

Hola queridos amigos (la editorial).

Antes que nada y por encima de todo, quiero felicitaros por vuestra revista, de la cual soy suscriptor, y que la encuentro muy interesante y de gran interés (pero los 2 primeros números me han llegado con un poco de retraso). Espero que los siguientes números sean del mismo estilo.

Después de felicitaros proseguiré con la carta ya que quiero haceros algunas preguntas y proposiciones.

a) Al recibir el primer número, correspondiente al mes de febrero, había en el sobre una etiqueta con mi nombre y con el número de socio (9.655). Y quisiera saber si este número sirve para algo, o simplemente es para saber la cantidad de socios que tenéis.

b) ¿No podríais publicar algún programa escrito ya en lenguaje máquina? Lo digo para no gastar memoria con el intérprete BASIC, y porque creo que puede ser una experiencia bastante interesante para los VIC-64-adictos.

c) Ahora, y por último, me gustaría hacer una proposición que viene debida por lo siguiente: El otro día expliqué a un amigo mío que la revista a la que yo estaba suscrito tenía más de 3.500 socios. Pero él no se lo creyó, y yo le dije que se lo demostraría como fuera.

Pues ahí va mi proposición.

Rogaría a todos los que leyeran esta carta que me enviaran una breve contestación en la cual pusiera el nº de socio y desde cuándo lo es.

Así pues, confiando en vuestras cartas y en vuestro espíritu de buenos socios, me despido.

Muchísimas gracias y animaos a escribir.

P.D.: Pasado un mes de la publicación de mi carta, escribiré una carta a la editorial para decirles los resultados.

XAVIER QUERAL CARRILLO
C/ CORCEGA, 613-5º 1
BARCELONA-25

Si y no. Ni tantos como 9.655 ni tan pocos como 3.500.

El número 9.655 no sabemos de qué puede ser, pero sospechamos que... ¡era el número de reembolso!

En este momento acabamos de dar el alta al número 4.146 y tu número en particular, Xavier, es el 1.011 —claro que los números sirven— el 5.000 visitará EE.UU. y el afortunado que le toque entre los demás, Inglaterra y las correspondientes instalaciones Commodore.

De todas formas nos has dado una idea. A partir del próximo número pondremos en las etiquetas el número de socio (para éste ya están hechas) para que todos puedan conocer el suyo.

En este número comenzamos un curso de lenguaje máquina, los programas comenzarán en un futuro.



MARKETCLUB

Servicio gratuito para nuestros lectores particulares. Empresas 300 ptas., por línea.

MERCADILLO

(Ref. M-1)

• CBM 4.032. Intercambio programas. José Marcé Mestres. Calle Sevilla, 5. Tel. (93) 803 77 51, de 8 a 3, **VILANOVA DEL CAM** (Barcelona).

(Ref. M-2)

• ACCESORIOS VIC-20: Ampliación de memoria 16K más varios programas, 13.000 Módulo de expansión para 6 cartuchos, 10.000. Cartucho lenguaje FORTH y manual, 7.000. Las tres cosas sólo por 25.000. Jaime. Tel.: 245 46 56. **BARCELONA.**

(Ref. M-3)

• Desearía vender por 40.000 ptas.: VIC-20, con cartucho Super Expander, las dos partes del Curso, el "joy-stick" y un juego Indescomp, todo comprado en diciembre del 82. Fernando Martínez, calle La Roda, 39, 5° D. Tel. 23 41 82. **ALBACETE.**

(Ref. M-4)

• Tengo un PET 2001/8K y desearía tener el cassette "Monitor para lenguaje Máquina". También desearía contactar con usuarios o clubs de PET si los hay. José Manuel Cámara Mas, calle Castor, 32, bloque II, 3º, puerta 1. **ALICANTE.**

(Ref. M-5)

• Se busca MODEM para Commodore 64. Precio a convenir. José M.ª Maci. Horno de Ladilla, 4, Montblanc. **TARRAGONA.** 86 00 63. Horas de comida.

(Ref. M-6)

• Vendo VIC-20 (Comprado en noviembre 1983), 2 cursos de basic, libros, programas, 4 cartuchos, juegos en cassette, cassette, joysticks. Equipo ideal para iniciarse en informática. Todo por 59.000 (Valor real 90.000). Llamar de 9 a 11 h. noche. Joan Sanz. Tel. 321 80 64. Travesera de las Corts, 295. **BARCELONA**, 29.

(Ref. M-7)

• Compraría Cassette y aplicación de memoria 8K para VIC-20. Precio a convenir. Manuel Aranda (435 49 00) Oficina; 91 (617 97 31) casa.

(Ref. M-8)

• Me urge la Unidad de Disco pero de 2.ª mano y daría hasta 40.000 ptas. El módulo "1020" lo tengo averiado, también lo compraría por 10.000 pesetas. José L. Castejón Roca - Las Paredes - **LA UNION.** Murcia.

(Ref. M-9)

• Desearía comprar o intercambiar programas de juegos o con diversas utilidades para el VIC-20 no ampliado (cinta). Carlos Plaza Hernández. Calle Mestre Nicolau. 37-41. Entlo. 3.ª. Teléfono: (93) 380 13 17. **BADALONA** (Barcelona). Preferible llamar de 13 a 16 ó de 22 a 23.

(Ref. M-10)

• Ruego toda información sobre programas y aplicaciones de 3032 así como monitor para lenguaje máquina con literatura en francés o español. Artemio González Pérez, Calle Fernando Arocena Quintero, 1, Barrio de Buenos Aires **SANTA CRUZ DE TENERIFE.** Teléfono: (922) 22 42 44.

(Ref. M-11)

• Vendo microordenador VIC-20 sin cassette, por 34.000 ptas. y regalo más de 70 programas de todo tipo, entre ellos, control stock, ajedrez (2), cuentas comerciales, archivo QSL con presentación en pantalla de la misma y posibilidad de imprimirla, juegos, etc. etc., Interface CW-RTTY autoconstruido, funciona a la perfección y vale para cualquier ordenador, se incluyen programas de CW y RTTY por 10.000 ptas. Manuel González Lombardía EA1CMC. C/Gabriel y Galán, 4-2ºA. **PEÑARANDA DE BARCAMONTE** (Salamanca).

(Ref. M-12)

• Intercambio programas de todo tipo para el VIC-20. Regalo programas a todo aquel que pida mi lista. También vendo ampliación de memoria 16K por sólo 12.000 pesetas. Dirigirse a: Tomás Toribio. Jerusalen, 5. **ZARAGOZA-9.** Teléfono: (976) 35 69 94.

(Ref. M-13)

Si os sobra una impresora para el VIC-20 y no sabéis a quién regalársela yo estaría muy complacido de recibirla. Jorge Rodríguez Añeyro, Polígono San Pablo, Barrio C, Torre 2º 4C. **SEVILLA-7.**

(Ref. M-14)

• Desearía comprar cartucho Super Expander para Vic-20. Precio a convenir. Llamar de 2 a 2.30 ó de 7 a 11. Tel. (954) 61 41 24. Calle: Gustavo Gallardo 7-4º D. **SEVILLA-13.** Cuanto antes llamen mejor.

CLUBS

(Ref. C-1)

• Club de amigos del VIC-20 y Commodore-64. Ramón P. Serna Soler. Foto Estudio 2. Plaza de Sombrereros, 2. **PALMA DE MALLORCA.** Tel.: 21 31 62.

(Ref. C-2).

• Desea información sobre clubs en existencia y gente que quiera formar uno. Agustín Quevedo Velasco. Oña. 55-4ºB. **MADRID-34.** Tel.: 202 94 28.

(Ref. C-3).

• Desea información sobre clubs en existencia y gente que quiera formar uno. Pablo Nistal Alonso. Empecinado, 46 6ºB. **MOSTOLES.** Tel.: 645 21 70.

(Ref. C-4).

• Club de programación Alaiz, en Pamplona. Está dirigido a estudiantes de B.U.P. y C.O.U. de esta ciudad, y el fin que tiene es la introducción en el mundo de la informática y programación BASIC. No tiene, por lo demás, ningún fin lucrativo. El club propiamente posee equipos COMMODORE 64, VIC-20 y diversos periféricos (impresora, unidad de disco y cassette). CLUB ALAIZ. Pza. Monasterio Santa Gemma, s/n. Tels.: 254480-257704. **PAMPLONA.**

(Ref. C-5).

• Club de Commodore de **Albacete.** Fernando Martínez Guerrero. La Roda, 39 5º D. **ALBACETE.**



(Ref. C-6).

• Círculo Durensan Vigues. **VIGO**. Venezuela, 48. Entrechan. Tel.: (986) 410683-422519.

(Ref. C-7).

• Un grupo de usuarios de la Comunidad Autónoma de Murcia intenta crear un Club de Usuarios de VIC-20 y C-64 en esta región, sin fines lucrativos. Interesados llamar a los teléfonos (968) 80 23 64 (Mañanas, preguntar por Paco); (968) 26 72 62 (Tardes, preguntar por Fernando). Francisco Sáez Soto. C/ Virgen del Rosario, 28. **ALCANTARILLA. MURCIA.**

SERVICIOS

(Ref. S-1)

• En Barcelona, clases de informática. PLAZAS LIMITADAS. Lenguaje BASIC. Prácticas con microordenador VIC-20. Prof. E. Martínez de Carvajal. Información: Tel. (93) 345 10 00. Señorita María José (mañanas) ó (93) 345 87 75.
Sr. Martínez (fuera de horas de oficina).

(Ref. S-2)

• Programación de ordenadores personales; organización explotación de ficheros; programas ordenadores auxiliares, para cuestiones empresariales, profesionales, administrativas, científicas. Mora Mas. Carlos III, 41. Tel.: 339 98 29 **BARCELONA-28.**

(Ref. S-3).

• Tengo programa para confección de documentos, cartas, textos, etc. Permite escribir lineas reales de impresora visualizando por pantalla y avisando al final de cada línea por timbre. Visualización del texto entero; modificaciones de línea; insertar lineas; grabación disco o cassette; lectura de datos de disco o cassette; control por pantalla del texto a rectificar; acepta; espacios, etc.; imprime normal y doble ancho; pregunta número de copias; salto automático de página en impresora; margen izquierdo en la página. Amadeo Bargay. Pza. Hospital 5-5º. Tel.: 874 41 93. **MANRESA** (Barcelona).

(Ref. S-4)

• PROGRAMADORES COMMODORE-64, excelente oportunidad para personas que dispongan de programas de gestión y utilidades y deseen comercializarlos en condiciones verdaderamente ventajosas (sólo programas de alta calidad). También programadores despuestos a realizar programas a medida. Deberán escribir al apartado de correos 24143 de Barcelona.

BOLSA DE TRABAJO

OFERTAS

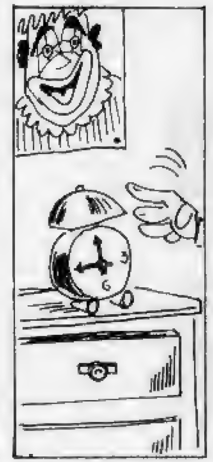
• TECNICO COMERCIAL, experiencia venta Microordenadores y/o material Auxiliar. Tel. (93) 207 39 12. Lunes 12. C. Mercantil.

• BECARIO con posibilidad de entrar en plantilla oficina MADRID. Requisitos: Licenciado o último curso; Inglés hablado y escrito; Interesado en software. Mandar "curriculum viate": APL *PLUS Service. Rosario Pino, 6. Madrid-20.

• Para la venta de material de uso informático se necesitan VENDEDORES. Se requiere: Coche propio; exento servicio militar; edad inferior a 27 años; formación grado medio. Incorporación inmediata. Interesados llamar al teléfono 457 49 23. Ref. SM-28/331101.

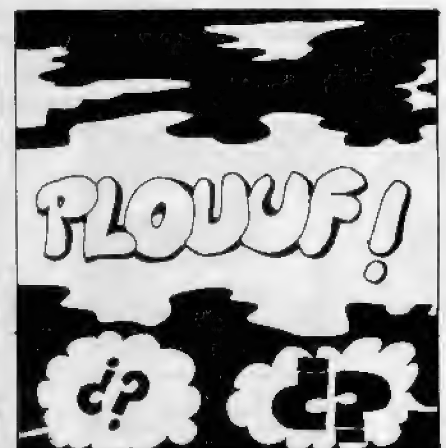

Las Aventuras de
RAMY ROMO
y sus amigos
por
CRRAZQUIN
Comienza la Aventura
UAAAAHH
¡VALE HOMBRE! ENSEGUIDA RECOJO. NO SE COMO PUEDES TENER SIEMPRE SUEÑO.


LO REALMENTE ESTUPENDO SERIA QUE ALGUIEN NOS CONTARA PASO A PASO TODOS LOS SECRETOS DE NUESTRO COMMODORE
¿NO CREES?
JUM
Z
Zz




CLIC!!
¿TE HAS LAVADO LOS DIENTES?
ZZZZ
BAH!


SSHHHHH
¿?


PLOUUF!
¿?


CLIC


¡HOLA MUCHACHOS! ¡NO OS ASUSTEIS! ME LLAMO RAFS...


¿RAFS? NO TE CONOCEMOS.. ¿A QUÉ HAS VENIDO?


VOSOTROS ME HABEIS LLAMADO...
¿NOSOTROOOS?


¡NO TE CONOCEMOS..!
..Y NADIE TE HA LLAMADO..
ESO
ADEMAS, MI CAMA ES MUY PEQUEÑA Y NO PUEDO ADMITIR A NADIE MAS..
¡BUENAS NOCHES!
TAP TAP TAP


¡ESCUCHADME ATENTAMENTE... SOY EL DUENDE RAFS O'RAFS Y ESTOY AQUI PARA LLEVAROS A CONOCER EL MUNDO MAGICO DEL COMMODORE!
¡¡!
... Y TODO ESO PORQUE VOSOTROS LO HABIAIS DESEADO HACE UNOS SEGUNDOS


JAJAJA
Ji Ji Ji


BIEN, LA UNICA FORMA DE QUE ME CREAIS ES LLEVAROS AL MUNDO DEL COMMODORE
POOUUUUFF


¡YUPP!!!! ES CIERTO ES UN DUENDE DE PRIMERAA!


AHHHJUMMM!!! AQUI ESTAMOS
¡¡FANTASTICO!! OYE FIJATE QUE ME PARECE QUE HASTA HAS CRECIDO
GLUPP!


¡NO! SOIS VOSOTROS LOS QUE OS HABEIS VUELTO MAS PEQUEÑOS..! ¡VAMOS! ..OS VOY A PRESENTAR A UNOS AMIGOS


A ESTAS HORAS SUELEN ESTAR REUNIDOS...
TOC TOC


AHH PERO SI ES RAFS O'RAFS
HOLA MR. CHIP
ERRAZQUIN/84


PASAD A TOMAR ALGO
VAMOS SO GALLINA MUEVE TEEE!
NOOOO SOCORROOO DEJAMEE!!


TE PRESENTO AL SR. VALVULEZ ...ES MUY SABIO Y TIENE MUCHOS AÑOS
HOLA ABUELO
¿QUE TAL?


¡ENCHANTÉ! SOY HENRI... VENGO DE ALBI, DE MUY LEJOS ...¿COMO ESTAIS?
ES UN PINTOR EXTRAORDINARIO
HOLA


AQUI ESTAN DOÑA TECLA, EL SEÑOR BYTE Y CONECTON


BROLMB !!
...Y ESE RU.. RU... RUIDO?


ES EL TERRIBLE EMPERADOR DE LA LOGICA, INTENTA SOMETER A TODO EL MUNDO A SU INMENSO PODER...


SI, MAS VALE QUE NUNCA OS ENCONTREIS CON EL O CON SUS SERVIDORES
Y HABLANDO DE COSAS ALEGRES,...
¿VOSOTROS COMO OS LLAMAIS?


¡HUMM!.. ES CIERTO AQUI DENTRO NO OS VALE VUESTRO NOMBRE DEL MUNDO EXTERIOR
HUY


ENTONCES HABRÁ QUE BUSCARNOS UN NUEVO NOMBRE..!


¡FANTASTICO! HAGAMOS UN CONCURSO ENTRE NUESTROS LECTORES
ESCRIBIDNOS A: PEDRO MUGURUZA, 4 - 8º B MADRID - 16 Y SORTEAREMOS ENTRE LAS CARTAS RECIBIDAS JUEGOS PARA VUESTRO COMMODORE
MANDA TUS NOMBRES PARA NOSOTROS Y PARTICIPA EN EL CONCURSO ... PUEDES GANAR UN JUEGO PARA TU COMMODORE!
ESCRIBID MUY RAPIDAMENTE!
AHH AHH
BZZZZ
¡HASTA EL PRÓXIMO NUMERO

El Manejo de Ficheros

Solucionando problemas con la Base de Datos "Deluxe"

Necesitabas un programa de base de datos barato para el VIC-20 o el C-64? Este no te va a costar un duro. Simplemente teclea el listado para almacenar, y clasificar tus datos con facilidad.



**VIC-20 con 3K o más.
C-64 con unidad
de discos 1540 ó 1541
ó Commodore Datassette**

El "Fichero Deluxe" es un manejador de ficheros para el C-64 o el VIC-20 con una ampliación de memoria de 3K o más. El programa está diseñado para uso con la unidad de discos simple 1540 ó 1541 o el Datassette de Commodore. Funcionará con cualquiera de las impresoras del VIC.

El formato del fichero es un grupo de páginas con diez entradas por páginas. En la versión para el VIC-20, la línea 2 comprueba la cantidad de memoria disponible para el almacenamiento de datos seleccionando el número ideal de páginas. Esto significa que si se cambia el tamaño del programa, esto será tenido en cuenta, y se modificará el número de páginas.

En la versión para el C-64, siempre se proporcionan 100 páginas, con diez entradas por página. En la línea 30, N se fija en 1000, es decir, el número de entradas que el fichero puede contener. Si quieres más o menos páginas, lo único que hay que hacer es cambiar este número.

Cuando se ejecuta el programa, lo primero que solicita es el nombre del fichero. Si se pulsa la tecla "Return" sin haber proporcionado un nombre, el nombre del fichero se fijará en "Sin Nombre". A continuación, el programa se inicializa.

Un momento más tarde, una lista de comandos de una letra aparecerán en pantalla. Los comandos son los siguientes: Página, Intercalar, Entrar, Catalogar, Clasificar, Borrar, New, Load (Cargar), Save (Salvar), Salida a Impresora y Ayuda. Si alguna vez se te olvida lo que significan, pulsa la tecla ? (Ayuda) para que salga una lista de las definiciones.

Utilizando los Comandos

Para llamar una página, se pulsa la P y el número de página deseada. (Un cursor intermitente le indicará al usuario que pulse la tecla "Return" después de teclear una información solicitada). Cuando la página aparece, se verán diez números de entrada seguidos por un guión. Para realizar una entrada, se pulsa la tecla E y se teclea uno de los números para indicar dónde tiene que colocarse la entrada. La entrada no puede contener comas, dos puntos, ni punto y coma. Después de haberse pulsado la tecla "Return", la entrada aparecerá en la página.

Si se desea que la entrada aparezca en el catálogo, tiene que ser invertida (las letras dentro de una barra de color). Para hacer esto, el primer carácter de la entrada tiene que ser una flecha a la izquierda. Esta es la tecla que se encuentra en la parte superior izquierda del teclado.

Normalmente, yo invierto los títulos de las listas. Por ejemplo, se podría clasificar una lista de libros según los nombres de los autores. Para hacer esto, se invierte el nombre de cada autor, entrando el título del libro detrás del nombre. (El título del libro **no** se invierte). Ahora, cuando el catálogo es llamado, se presentará el nombre del autor junto con el número de página donde aparece.

Para insertar algo entre dos entradas existentes, se pulsa I y se teclea el número de línea donde se va a colocar la inserción. Para borrar una entrada, se pulsa la tecla Q y se teclea el número de entrada. Para cancelar un comando como Borrar, se teclea la letra de otro comando en vez del número de entrada.

Si el usuario quiere repasar las páginas rápidamente, se pulsa la barra de espaciado en vez de la letra de un comando. Para llamar el catálogo, se pulsa la C. Debido al espacio limitado en la pantalla, sólo diez líneas con sus respectivos números de página pueden ser representadas a la vez. Para que aparezcan las diez líneas siguientes, se pulsa la tecla "Espacio".

Para salvar o cargar un fichero, se pulsan las teclas S o L respectivamente. El ordenador le solicitará al usuario que confirme lo pedido. Esto asegura que no se cargue un fichero antes de terminar con el que está en el ordenador. A continuación, se le solicitará al usuario que indique si está usando una cinta o una unidad de

discos. Se pulsa la D para disco y la C para cinta. Si se pulsa la D, el ordenador preguntará si el usuario desea un listado de los ficheros de datos salvados en disco.

Cuando se vuelve a salvar un fichero en disco, el fichero anterior será sustituido por el nuevo. De esta forma, no hace falta recordar cuál es la versión más reciente de un fichero.

Para clasificar, se pulsa la tecla A. Sólo se puede clasificar lo que se incluya dentro de una lista. Si las listas son los nombres de autores, los libros pueden ser clasificados según cada autor individual, pero los nombres mismos de los autores no pueden clasificarse. Después de pulsar la A, el ordenador solicitará el número de la primera entrada para clasificarse.

BASE DE DATOS VIC-20

```
1 T$="SIN[SPC]NOMBRE":PRINT"[CLR]NOM
BRE[SPC]DE[SPC]ARCHIVO":INPUTT$:T$=L
EFT$(T$,14)
2 N=INT(FRE(0)/20):P=1:X=(N+1)/10:DI
MS$(N):POKE36879,185:R$="[SPC]LINEA#
":GOSUB81
3 F$=CHR$(15):G$=CHR$(8):GOTO8
4 K=0
5 Q=0:PRINT"[CLR][WHT][RVSON]CATALOG
O:[SPC]";T$:PRINT"PAGINA"
6 FORJ=KTON:IFASC(S$(J))=18THENPRINT
"[CRSRL]"INT(J/10+1);S$(J):Q=Q+1:IFQ
>10THEN8
7 NEXTJ
8 GOSUB26:IFA=0THEN8
9 ONAGOTO4,12,19,22,29,36,41,67,47,5
2,31
10 IFJ<N+1THENK=J:GOTO5
11 GOTO4
12 PRINT"[2CRSRD]PAGINA#":INPUTA$:P=
VAL(A$):GOSUB27:IFA<>0THEN9
13 IFP<1ORP>XTHENPRINT"[3CRSRU]":GOT
O12
14 PRINT"[RVSON][CLR]PAGINA"P;T$:FOR
I=0TO9:L=(P-1)*10+I:PRINT"[CRSRL]"L;
S$(L):NEXT
15 GOSUB26:IFA=0THEN15
16 IFA<>12THEN9
17 P=P+1:IFP>XTHENP=1
18 GOTO14
19 A$="-1":PRINT"[2CRSRD]ENTRAR"R$:I
NPUTA$:J=VAL(A$):GOSUB27:IFA<>0THEN9

20 IFJ<0ORJ>INT(X)*10+1THENPRINT"[5CRSRU]
":GOTO19
21 INPUTS$(J):GOTO116
22 A$="-1":PRINT"[2CRSRD]INTERCALAR"
R$:INPUTA$:J=VAL(A$):GOSUB27:IFA<>0T
HEN9
23 IFJ<0ORJ>INT(X)*10+1THENPRINT"[5CRSRU]
":GOTO22
24 PRINT"ENTRADA":INPUTD$:PRINT"INSE
RTANDO":IFR=NTHEN14
25 GOSUB86:FORI=KKTOJ+1STEP-1:S$(I)=
S$(I-1):NEXT:S$(J)=D$:GOTO116
26 E$="C,P,E,I,N,S,L,H,Q,A,?[RVSOFF]
[SPC]":PRINT"[RVSON][CRSRD]"E$"[CRSRL]
[2CRSRU]":GOSUB63
27 A=0:FORI=1TO13:IFMID$(E$,I*2-1,1)
=A$THENA=I:I=13
28 NEXTI:RETURN
29 GOSUB66:IFA$<>"S"THEN4
30 PRINT"NUEVO[SPC]NOMBRE[SPC]DE[SPC]
ARCHIVO":INPUTT$:GOSUB81:GOTO4
31 PRINT"[CLR][RVSON]C[RVSOFF]ATALOG
O[CRSRD]":PRINT"VER[SPC][RVSON]P[RVSOFF]
AGINA[CRSRD]":PRINT"[RVSON]I[RVSOFF]
NTERCALAR"R$
32 PRINT"[CRSRD][RVSON]E[RVSOFF]NTRA
R"R$:PRINT"[CRSRD][RVSON]Q[RVSOFF]UI
TAR"R$:PRINT"[CRSRD]ORDEN[SPC][RVSON]
A[RVSOFF]LFABETICO[CRSRD]":PRINT"[RVSON]
N[RVSOFF]UEVO[SPC]ARCHIVO[CRSRD]"
33 PRINT"[RVSON]L[RVSOFF]OAD[SPC]DE[SPC]
CINTA[SPC]O[SPC]DISCO":PRINT"[CRSRD]
[RVSON]S[RVSOFF]AVE[SPC]A[SPC]CINTA[SPC]
O[SPC]DISCO[CRSRD]"
34 PRINT"[RVSON]?[RVSOFF][SPC]DEFINI
CIONES":PRINT"[CRSRD][RVSON]H[RVSOFF]
[SPC]LISTAR[SPC]A[SPC]IMPRESORA[CRSRU]
":GOSUB26:IFA=0THEN31
35 GOTO9
36 PRINT"[CLR][BLK]SAVE[SPC]A[SPC]CI
NTA[SPC]O[SPC]DISCO":GOSUB66:IFA$<>"
S"THEN4
37 PRINT"[RVSON]D[RVSOFF]ISCO[SPC]O[SPC]
[RVSON]C[RVSOFF]INTA?":GOSUB63:IFA$=
"C"THEN4
38 IFA$<>"D"THEN4
39 OPEN15,8,15,"UI-":CLOSE15:OPEN2,8
,2,"@0:"+T$+"[SPC][SHIFTX],S,W":J=2:
GOTO89
40 OPEN1,1,1,T$:PRINT#1,T$:J=1:GOTO8
9
41 PRINT"[CLR][RVSON]LOAD[SPC]DE[SPC]
CINTA[SPC]O[SPC]DISCO":GOSUB66:IFA$<
>"S"THEN4
42 PRINT"[RVSON]D[RVSOFF]ISCO[SPC]O[SPC]
[RVSON]C[RVSOFF]INTA?":GOSUB63:IFA$=
"C"THEN46
43 IFA$<>"D"THEN4
44 GOSUB97:PRINT"[CRSRD]NOMBRE[SPC]D
E[SPC]ARCHIVO?":INPUTN$:IFN$=""THEN4

45 OPEN15,8,15,"UI-":CLOSE15:OPEN2,8
,2,"0:"+N$+"[SPC][SHIFTX],S,R":J=2:G
OTO93
46 OPEN1,1,0:J=1:GOTO93
47 PRINT"[2CRSRD]QUITAR[SPC]LA[SPC]L
INEA"R$:INPUTA$:J=VAL(A$):GOSUB27:IF
A<>0THEN9
48 IFJ<0ORJ>NTHENPRINT"[2CRSRU]":GOT
O47
49 FORI=JTON-1:IFS$(I)="[SHIFT*]"AND
S$(I+1)="[SHIFT*]"THENI=N-1:GOTO51
50 S$(I)=S$(I+1):GOSUB65
51 NEXT:S$(N)="[SHIFT*]":GOTO14
52 PRINT"[2CRSRD]ORDENAR[SPC]";:GOSU
B82:U=VAL(A$):IFU<0ORU>NTHENGOTO8
53 IFMID$(S$(U),1,1)="[RVSON]"THENU=
U+1
54 GOSUB83:K=0:FORI=UTON:IFLEFT$((S
$(I)),1)="[RVSON]"THENK=1:NN=I-1:I=K
K
55 NEXTI:IFK=1THEN57
56 NN=K
57 I=0
58 J=U:IFI=NN-UTHEN14
59 IFJ=NN-ITHEN62
60 IFS$(J)>S$(J+1)THENP$=S$(J):S$(J)
=S$(J+1):S$(J+1)=P$
61 J=J+1:GOTO59
62 I=I+1:POKE36879,INT(RND(K)*8+184)
:GOTO58
63 GETA$:IFA$=""THEN63
64 RETURN
65 PRINT"[HOM]"TAB(37);I:RETURN
66 PRINT"[2CRSRD][RVSON]ESTAS[SPC]SE
GURO?[SPC]S/N":GOSUB63:RETURN
67 PRINT"[CLR][RVSON]COPIA[SPC]IMPRE
SA":GOSUB66:IFA$<>"S"THEN4
68 PRINT"EL[SPC]ARCHIVO[SPC]ENTERO?"
:GOSUB63:IFA$="S"THENF=0:M=N:GOTO71
69 GOSUB82:F=VAL(A$):IFF<0ORF>NGOTO6
7
70 PRINT"TERMINANDO[SPC]EN"R$:INPUTA
$:M=VAL(A$):IFM<0ORM>NORM=<FTHEN67
```

Sigue

LISTADO 1 (Cont.)

```
71 OPEN4,4:A$="":FORI=1TOINT((40-LEN
(T$))/2)
72 A$=A$+"[SPC]":NEXTI:A$=A$+"[RVSOFF]
":PRINT#4,CHR$(14)A$+T$+F$CHR$(10)CH
R$(10)
73 LL=4:FORI=FTOM:IFLEFT$(S$(I),1)="
[RVSON]"THEN76
74 IFS$(I)<>"[SHIFT*]"THENPRINT#4,CH
R$(15)"[5SPC]"S$(I)CHR$(10):LL=LL+2
75 GOTO79
76 A$="":FORKK=7TOLEN(S$(I))*6:A$=A$
+CHR$(255):NEXTKK
77 PRINT#4,"[5SPC]"G$+A$:PRINT#4,F$+
"[5SPC]"+S$(I)+G$
78 PRINT#4,F$+"[5SPC]"+G$+A$+F$CHR$(
10):LL=LL+4
79 IFLL>59THENFORNL=LLTO72:PRINT#4,C
HR$(10);:NEXTNL:PRINT#4,"":LL=3
80 NEXTI:CLOSE4:GOTO4
81 FORJ=0TON:S$(J)="[SHIFT*]":NEXTJ:
RETURN
82 PRINT"COMENZANDO[SPC]EN"R$:INPUTA
$:RETURN
83 FORI=NTO0STEP-1:IFS$(I)<>"[SHIFT*]
"THENKK=I:I=0:GOTO85
84 KK=I
85 NEXTI:RETURN
86 FORI=J+1TON:IFS$(I)="[SHIFT*]"THE
NKK=I:I=N:GOTO85
87 KK=I
88 NEXTI:RETURN
89 FORI=NTO0STEP-1:K=I:IFS$(I)<>"[SHIFT*]
"THENI=0
90 NEXTI:K=K+1
91 PRINT#J,T$+"[SPC][SHIFTX]":PRINT#
J,K:FORI=0TOK
92 PRINT#J,S$(I):GOSUB65:NEXT:CLOSEJ
:GOTO4
93 INPUT#J,N$:IFRIGHT$(N$,2)<>"[SPC]
[SHIFTX]"THENPRINTN$"[SPC]NO[SPC]ENC
ONTRADO":GOSUB63:CLOSEJ:GOTO4
94 T$=N$:T$=LEFT$(T$,LEN(T$)-2):PRIN
T"[CRSRD][RVSON]ENCONTRADO[SPC]"T$:I
NPUT#J,K
95 IFK>NTHENPRINT"[CRSRD][RVSON]ATEN
CION:":PRINT"EL[SPC]ARCHIVO[SPC]FUE[SPC]
CREADO[SPC]CON[SPC]MAS[SPC]MEMORIA"
K=N
96 FORI=0TOK:INPUT#J,S$(I):GOSUB65:N
EXT:CLOSEJ:GOTO4
97 PRINT"[RVSON][CLR]LISTA[SPC]DE[SPC]
LOS[SPC]FICHEROS[SPC]EN[SPC]DISCO?[CRSRD]
":GOSUB63:IFA$<>"S"THENRETURN
98 PRINT"[CLR][RVSON]BUSCANDO[SPC]EN
[SPC]EL[SPC]DISCO[CRSRD]"
99 OPEN1,8,0,"$0"
100 GET#1,A$,B$
101 GET#1,A$,B$
102 GET#1,A$,B$
103 C=0
104 IFA$<>""THENC=ASC(A$)
105 IFB$<>""THENC=C+ASC(B$)*256
106 D$=D$+"[RVSON]"+MID$(STR$(C),2)+
"[RVSOFF]"
107 GET#1,B$:IFST<>0THENCLOSE1:PRINT
:RETURN
108 IFB$<>CHR$(34)THEN107
109 GET#1,B$:IFB$<>CHR$(34)THEND$=D$
+B$:GOTO109
110 GET#1,B$:IFB$=CHR$(32)THEN110
111 C$=""
112 C$=C$+B$:GET#1,B$:IFB$<>""THEN11
2
113 IFRIGHT$(D$,1)="[SHIFTX]"THENPRI
NTD$
114 D$="":GOTO101
115 Y=Y-1:GOTO118
116 IFLEFT$(S$(J),1)="←"THENS$(J)="[RVSON]
"+RIGHT$(S$(J),LEN(S$(J))-1)
117 GOTO14
118 FORI=0TOC:C$(I,P%(L))="[SHIFT*]"
:NEXTI:FORI=LTOY:P%(I)=P%(I+1):NEXTI
:L=L-1:RETURN
```

LISTADO 2. BASE DE DATOS C-64

```
10 POKE53281,14:T$="SIN[SPC]NOMBRE"
20 PRINT"[CLR][WHT]NOMBRE[SPC]DE[SPC]
ARCHIVO":INPUTT$
25 PRINT"[6CRSRD][14SPC]ESPERE[SPC]P
OR[SPC]FAVOR":T$=LEFT$(T$,14)
30 N=1000:P=1:X=(N+1)/10:DIMS$(N):R$
="[SPC]LINEA#":GOSUB810:GOTO80
40 K=0
50 Q=0:PRINT"[CLR][WHT][RVSON]CATALO
GO:[SPC]";T$"[2CRSRD]":PRINT"PAGINA"

60 FORJ=KTON:IFASC(S$(J))=18THENPRIN
T"[CRSRL]"INT(J/10+1);S$(J):Q=Q+1:IF
Q>10THEN80
70 NEXTJ
80 GOSUB260:IFA=0THEN80
90 ONAGOTO40,120,190,220,290,360,410
,670,470,520,310
100 IFJ<N+1THENK=J:GOTO50
110 GOTO40
120 PRINT"[2CRSRD]PAGINA#":INPUTA$:P
=VAL(A$):GOSUB270:IFA<>0THEN90
130 IFP<1ORP>XTHENPRINT"[3CRSRU]":GO
TO120
140 PRINT"[RVSON][CLR]PAGINA"P;T$"[2CRSRD]
":FORI=0TO9:L=(P-1)*10+I:PRINT"[CRSRL]
"L;S$(L):NEXT
150 GOSUB260:IFA=0THEN150
160 IFA<>12THEN90
170 P=P+1:IFP>XTHENP=1
180 GOTO140
190 A$="-1":PRINT"[2CRSRD]ENTRAR"R$:
INPUTA$:J=VAL(A$):GOSUB270:IFA<>0THE
N90
200 IFJ<0ORJ>INT(X)*10+1THENPRINT"[5CRSRU]
":GOTO190
210 INPUTS$(J):GOTO1160
220 A$="-1":PRINT"[2CRSRD]INTERCALAR
"R$:INPUTA$:J=VAL(A$):GOSUB270:IFA<>
0THEN90
230 IFJ<0ORJ>INT(X)*10+1THENPRINT"[5CRSRU]
":GOTO220
240 PRINT"ENTRADA":INPUTD$:PRINT"INS
ERTANDO":IFR=NTHEN140
250 GOSUB860:FORI=KKTOJ+1STEP-1:S$(I
)=S$(I-1):NEXT:S$(J)=D$:GOTO1160
260 E$="C,P,E,I,N,S,L,H,Q,A,?[RVSOFF]
[SPC]":PRINTTAB(9)"[RVSON][CRSRD]"E$
"[CRSRL][2CRSRU]":GOSUB630
270 A=0:FORI=1TO13:IFMID$(E$,I*2-1,1
)=A$THENA=I:I=13
280 NEXTI:RETURN
290 GOSUB660:IFA$<>"S"THEN40
300 PRINT"NUEVO[SPC]NOMBRE[SPC]DE[SPC]
ARCHIVO":INPUTT$:GOSUB810:GOTO40
310 PRINT"[CLR][RVSON]C[RVSOFF]ATALO
GO[CRSRD]":PRINT"VER[SPC][RVSON]P[RVSOFF]
AGINA[CRSRD]":PRINT"[RVSON]I[RVSOFF]
NTERCALAR"R$
320 PRINT"[CRSRD][RVSON]E[RVSOFF]NTR
AR"R$:PRINT"[CRSRD][RVSON]Q[RVSOFF]U
ITAR"R$:PRINT"[CRSRD]ORDEN[SPC][RVSON]
```

```
A[RVSOFF]LFABETICO[CRSRD]"
325 PRINT"[RVSON]N[RVSOFF]UEVO[SPC]A
RCHIVO[CRSRD]"
330 PRINT"[RVSON]L[RVSOFF]OAD[SPC]DE
[SPC]CINTA[SPC]O[SPC]DISCO":PRINT"[CRSRD]
[RVSON]S[RVSOFF]AVE[SPC]A[SPC]CINTA[SPC]
O[SPC]DISCO[CRSRD]"
340 PRINT"[RVSON]?[RVSOFF][SPC]DEFIN
ICIONES":PRINT"[CRSRD][RVSON]H[RVSOFF]
[SPC]LISTAR[SPC]A[SPC]IMPRESORA[CRSRU]
":GOSUB260:IFA=0THEN310
350 GOTO90
360 PRINT"[CLR][BLK]SAVE[SPC]A[SPC]C
INTA[SPC]O[SPC]DISCO":GOSUB660:IFA$<
>"S"THEN40
370 PRINT"[RVSON]D[RVSOFF]ISCO[SPC]O
[SPC][RVSON]C[RVSOFF]INTA?":GOSUB630
:IFA$="C"THEN400
380 IFA$<>"D"THEN40
390 OPEN2,8,2,"@0:"+T$+"[SPC][SHIFTX]
,S,W":J=2:GOTO890
400 OPEN1,1,1,T$:PRINT#1,T$:J=1:GOTO
890
410 PRINT"[CLR][RVSON]LOAD[SPC]DE[SPC]
CINTA[SPC]O[SPC]DISCO":GOSUB660:IFA$
<>"S"THEN40
420 PRINT"[RVSON]D[RVSOFF]ISCO[SPC]O
[SPC][RVSON]C[RVSOFF]INTA?":GOSUB630
:IFA$="C"THEN460
430 IFA$<>"D"THEN40
440 GOSUB970:PRINT"[CRSRD]NOMBRE[SPC]
DE[SPC]ARCHIVO?":INPUTN$:IFN$=""THEN
40
450 OPEN2,8,2,"0:"+N$+"[SPC][SHIFTX]
,S,R":J=2:GOTO930
460 OPEN1,1,0:J=1:GOTO930
470 PRINT"[2CRSRD]QUITAR[SPC]LA[SPC]
LINEA"R$:INPUTA$:J=VAL(A$):GOSUB270:
IFA<>0THEN90
480 IFJ<0ORJ>NTHENPRINT"[2CRSRU]":GO
TO470
490 FORI=JTON-1:IFS$(I)="[SHIFT*]"AN
DS$(I+1)="[SHIFT*]"THENI=N-1:GOTO510

500 S$(I)=S$(I+1):GOSUB650
510 NEXT:S$(N)="[SHIFT*]":GOTO140
520 PRINT"[2CRSRD]ORDENAR[SPC]":GOS
UB820:U=VAL(A$):IFU<0ORU>NTHENGOTO80

530 IFMID$(S$(U),1,1)="[RVSON]"THENU
=U+1

540 GOSUB830:K=0:FORI=UTOKK:IFLEFT$(
(S$(I)),1)="[RVSON]"THENK=1:NN=I-1:I
=KK
550 NEXTI:IFK=1THEN570
560 NN=KK
570 I=0
580 J=U:IFI=NN-UTHEN140
590 IFJ=NN-ITHEN620
600 IFS$(J)>S$(J+1)THENP$=S$(J):S$(J
)=S$(J+1):S$(J+1)=P$
610 J=J+1:GOTO590
620 I=I+1:POKE53280,INT(RND(K)*15+1)
:GOTO580
630 GETA$:IFA$=""THEN630
640 RETURN
650 PRINT"[HOM]"TAB(30),I:RETURN
660 PRINT"[2CRSRD][RVSON]ESTAS[SPC]S
EGURO?[SPC]S/N":GOSUB630:RETURN
670 PRINT"[CLR][RVSON]COPIA[SPC]IMPR
ESA":GOSUB660:IFA$<>"S"THEN40
680 PRINT"EL[SPC]ARCHIVO[SPC]ENTERO?
":GOSUB630:IFA$="S"THENF=0:M=N:GOTO7
10
690 GOSUB820:F=VAL(A$):IFF<0ORF>NGOT
O670
700 PRINT"TERMINANDO[SPC]EN"R$:INPUT
A$:M=VAL(A$):IFM<0ORM>NORM=<FTHEN670

710 OPEN4,4:A$="":FORI=1TOINT((40-LE
N(T$))/2)
720 A$=A$+"[SPC]":NEXTI:A$=A$+"[RVSOFF]
":PRINT#4,CHR$(14)A$+T$+F$CHR$(10)CH R$(10)
730 LL=4:FORI=FTOM:IFLEFT$(S$(I),1)=
"[RVSON]"THEN760
740 IFS$(I)<>"[SHIFT*]"THENPRINT#4,C
HR$(15)"[5SPC]"S$(I)CHR$(10):LL=LL+2

750 GOTO790
760 A$="":FORKK=7TOLEN(S$(I))*6:A$=A
$+CHR$(255):NEXTKK
770 PRINT#4,"[5SPC]"G$+A$:PRINT#4,F$
+"[5SPC]"+S$(I)+G$
780 PRINT#4,F$+"[5SPC]"+G$+A$+F$CHR$
(10):LL=LL+4
790 IFLL>59THENFORNL=LLTO72:PRINT#4,
CHR$(10);:NEXTNL:PRINT#4,"":LL=3
800 NEXTI:CLOSE4:GOTO40
810 FORJ=0TON:S$(J)="[SHIFT*]":NEXTJ
:RETURN
820 PRINT"COMENZANDO[SPC]EN"R$:INPUT
A$:RETURN
830 FORI=NTO0STEP-1:IFS$(I)<>"[SHIFT*]
"THENKK=I:I=0:GOTO850
840 KK=I
850 NEXTI:RETURN
860 FORI=J+1TON:IFS$(I)="[SHIFT*]"TH
ENKK=I:I=N:GOTO850
870 KK=I
880 NEXTI:RETURN
890 FORI=NTO0STEP-1:K=I:IFS$(I)<>"[SHIFT*]
"THENI=0
900 NEXTI:K=K+1
910 PRINT#J,T$+"[SPC][SHIFTX]":PRINT
#J,K:FORI=0TOK
920 PRINT#J,S$(I):GOSUB650:NEXT:CLOS
EJ:GOTO40
930 INPUT#J,N$
935 IFRIGHT$(N$,2)<>"[SPC][SHIFTX]"T
HENPRINTN$"[SPC]NO[SPC]ENCONTRADO":G
OSUB630:CLOSEJ:GOTO40
940 T$=N$:T$=LEFT$(T$,LEN(T$)-2):PRI
NT"[CRSRD][RVSON]ENCONTRADO[SPC]"T$:
INPUT#J,K
950 IFK>NTHENPRINT"[CRSRD][RVSON]ATE
NCION:":PRINT"EL[SPC]ARCHIVO[SPC]FUE
[SPC]CREADO[SPC]CON[SPC]MAS[SPC]MEMO
RIA":K=N
960 FORI=0TOK:INPUT#J,S$(I):GOSUB650
:NEXT:CLOSEJ:GOTO40
970 PRINT"[RVSON][CLR]LISTA[SPC]DE[SPC]
LOS[SPC]FICHEROS[SPC]EN[SPC]DISCO?[CRSRD]
":GOSUB630:IFA$<>"S"THENRETURN
980 PRINT"[CLR][RVSON]BUSCANDO[SPC]E
N[SPC]EL[SPC]DISCO[CRSRD]"
990 OPEN1,8,0,"$0"
1000 GET#1,A$,B$
1010 GET#1,A$,B$
1020 GET#1,A$,B$
1030 C=0
1040 IFA$<>""THENC=ASC(A$)
1050 IFB$<>""THENC=C+ASC(B$)*256
1060 D$=D$+"[RVSON]"+MID$(STR$(C),2)
+"[RVSOFF]"
1070 GET#1,B$:IFST<>0THENCLOSE1:PRIN
T:RETURN
1080 IFB$<>CHR$(34)THEN1070
1090 GET#1,B$:IFB$<>CHR$(34)THEND$=D
$+B$:GOTO1090
1100 GET#1,B$:IFB$=CHR$(32)THEN1100
1110 C$=""
1120 C$=C$+B$:GET#1,B$:IFB$<>""THEN1
120
1130 IFRIGHT$(D$,1)="[SHIFTX]"THENPR
INTD$
1140 D$="":GOTO1010
1150 Y=Y-1:FORI=0TOL:C$(1,P%(L))="[SHIFT*]
":NEXTI:FORI=LTOY:P%(I)=P%(I+1):NEXT
I
1155 L=L-1:RETURN
1160 IFLEFT$(S$(J),1)="←"THENS$(J)="
[RVSON]"+RIGHT$(S$(J),LEN(S$(J))-1
1170 GOTO140
```

EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina

CAPITULO I

Por Diego ROMERO

Introducción

Dado el creciente interés que está despertando entre nuestros lectores la programación en lenguaje máquina y la escasez de publicaciones en español sobre este tema, hemos pensado comenzar en este número un cursillo sobre esta materia que empiece desde cero para que todos puedan seguirlo.

Aquellos que ya tengan conocimientos sobre este tema comprenderán que es necesario partir de cero en un curso de este tipo, y espero que tengan paciencia hasta que lleguemos a un nivel que pueda ser interesante para ellos, aunque no estaría nada mal que repasasen un poco lo que ya saben desde hace tiempo... y de paso podrían hacerme llegar cartas con opiniones y sugerencias sobre el tema.

Existe una primera dificultad a la hora de programar en código máquina, y es que el ordenador trabaja con ceros y unos, mientras que nosotros lo hacemos con números del cero al nueve. Para tratar de aclarar un poco el asunto, aquí tenéis este artículo y el programa que lo acompaña.

Sistemas de Numeración

Desde que el hombre existe, creo que decidió utilizar los dedos para contar cuántas gallinas le daría a su vecino a cambio de su cerdo, de aquí que comenzase a hacer sus cálculos con diez números del cero al nueve; y si no fijaros en los niños pequeños que todavía no saben contar... ¿Cómo lo hacen?... Con los dedos.

Nota: Esta es una opinión muy personal y puede no coincidir con la realidad. De hecho creo que no fueron gallinas, me dijeron que pudieron ser terodáctilos prehistóricos o algo similar.

Sea como fuere, el caso es que nosotros utilizamos el sistema de numeración llamado decimal, y los ordenadores trabajan con ceros y unos solamente, decimos que trabajan en binario. ¿Cómo podremos descifrar este sistema de numeración?...

Vamos a fijarnos un poco en el nuestro. Tomamos un número como ejemplo y lo descomponemos:

$709=7\times10^2+0\times10^1+9\times10^0=700+0+9$

¿Cómo lo haríamos con el número 1101 de nuestro ordenador? Veamos...

$1101=1\times2^3+1\times2^2+0\times2^1+1\times2^0=8+4+0+1=13$

Aquí tenemos la norma para realizar la conversión... Para pasar un número de binario a decimal basta con ir sumando el número 2 elevado a la potencia correspondiente al lugar que ocupe cada dígito uno en el número. Los lugares ocupados se numeran de derecha a izquierda comenzando por cero:

2^N, ..., 2^4, 2^3, 2^2, 2^1, 2^0.

luego 1 0 1 1 1 0 será igual a:

$2^5+0+2^3+2^2+2^1+0 = 32+8+4+2=46$

Para realizar la conversión en sentido inverso, tenemos que ir dividiendo por dos el número que queremos pasar a binario, y los restos de las sucesivas divisiones junto con el último cociente nos dan el número binario buscado.

Ejemplo: 23 : 2 = 11 resto 1
11 : 2 = 5 resto 1
5 : 2 = 2 resto 1
2 : 2 = 1 resto 0

Como el último cociente ha sido 1, entonces 23 en decimal equivale a 10111 en binario.

Otro ejemplo:
10 : 2 = 5 resto 0
5 : 2 = 2 resto 1
2 : 2 = 1 resto 0

El último cociente fue 1, luego 10 decimal es 1010 en binario.

Ya hemos visto cómo pasar un número de binario a decimal y viceversa. Pero tenemos un problema más, que es la facilidad de equivocarnos que podemos tener cuando escribimos un número con varios unos y ceros. Para evitar esto los agruparemos en grupos de cuatro bits (dígitos binarios) y los utilizaremos en hexadeci-

Para
VIC-20 y C-64

BINARIO	HEXADECIMAL	DECIMAL	OCTAL
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	10
1001	9	9	11
1010	A	10	12
1011	B	11	13
1100	C	12	14
1101	D	13	15
1110	E	14	16
1111	F	15	17

mal (base 16). La equivalencia entre binario y hexadecimal la podéis ver en la siguiente tabla.

Antes vimos que el 23 decimal era 10111 en binario, y en hexadecimal será 17 (0001 0111).

Como podéis observar ha aparecido una nueva columna en la tabla, la del OCTAL, que es un sistema de numeración con base 8. El cambio de base octal a decimal se hace igual que el del binario a decimal, pero usando las potencias de 8 en lugar de las de 2.

Ejemplo:
27 en octal es $2\times8^1+7\times8^0=16+7=23$ en decimal.

El cambio en sentido inverso es igual pero dividiendo por 8 en lugar de dividir por 2.

Para efectuar la conversión de decimal a hexadecimal, es mejor pasarlo a binario y agruparlo en grupos de cuatro bits.

Ejemplo: 84 decimal será en binario

84 : 2 = 42 resto 0
42 : 2 = 21 resto 0
21 : 2 = 10 resto 1
10 : 2 = 5 resto 0
5 : 2 = 2 resto 1
2 : 2 = 1 resto 0

Ultimo cociente = 1 ... en binario es 1010100

Si el número binario 1010100 lo agrupamos en grupos de cuatro bits, será 101 0100 y mirando la tabla vemos que corresponde a... 5 4 en hexadecimal.

En octal las divisiones por 8 nos darían:
84 : 8 = 10 resto 4
10 : 8 = 1 resto 2

Ultimo cociente = 1 luego en octal sería 124.

Otro modo de obtener este resultado podría haber sido agrupando el número binario en grupos de tres bits y mirando en la tabla.

1010100 = 001 010 100 = 1 2 4 en octal.

¿Cómo distinguir un número cuando está en base 2, 8, 10 ó 16?

Teóricamente se debe indicar la base en que está escrito un número añadiéndole un subíndice a la derecha, pero ésta notación complicaría demasiado la escritura de números.

Ejemplo: $84_{(10} = 124_{(8} = 54_{(16} = 1010100_{(2}$

Para simplificar la notación, se suele utilizar el símbolo del dolar ($) para indicar que el número es hexadecimal, el de tanto por ciento (%) para el binario, y el símbolo @ para el octal. Dejando los números decimales sin prefijo.

Ejemplo: 84 = @ 124 = $54 = %1010100

Para practicar sobre la conversión de una base a otra, conviene que lo hagáis con papel y lápiz hasta comprender bien la mecánica del cambio de base, y podéis utilizar el programa que aparece en el listado para comprobar que no habéis cometido errores en la conversión.

El programa pregunta la base en que queréis introducir el número y contestará el mismo número en las otras bases.

En el próximo número comenzaremos con la descripción de la serie de instrucciones básicas de carga y almacenamiento de un registro, es decir, con el principio de lo dificilillo.

¡Practicad la conversión! y...

Hasta pronto.

```
1 REM* PROGRAMA CAMBIOS DE BASE *
2 PRINT"[CLR][CRSRD][3SPC]CAMBIO[SPC]
DE[SPC]BASE"
3 PRINT"[2CRSRD][RVSON]1[RVSOFF][SPC]
BINARIO"
4 PRINT"[2CRSRD][RVSON]2[RVSOFF][SPC]
OCTAL"
5 PRINT"[2CRSRD][RVSON]3[RVSOFF][SPC]
DECIMAL"
6 PRINT"[2CRSRD][RVSON]4[RVSOFF][SPC]
HEXADECIMAL"
7 ME$="[2SPC]DATO[SPC](1/4)?":GOSUB2
6
8 PRINT"[CRSRU][21SPC][CRSRU]":ONVAL
(Z$)GOTO10,12,14,16
9 GOTO7
10 BASE=2:INPUT"BIN";DA$:GOSUB31:IFE
R=1THEN10
11 GOTO17
12 BASE=8:INPUT"OCT";DA$:GOSUB31:IFE
R=1THEN12
13 GOTO17
14 BASE=10:INPUT"DEC";DA$:GOSUB31:IF
ER=1THEN14
15 GOTO17
16 BASE=16:INPUT"HEX";DA$:GOSUB31:IF
ER=1THEN16
17 IFNR<2↑20THEN19
18 PRINT:PRINT:ME$="DEMASIADO[SPC]GR
ANDE":GOSUB26:GOTO1
19 PRINT"[CLR][CRSRD][SPC]EQUIVALENT
ES:"
20 PRINT"[2CRSRD]BIN=[SPC]%[SPC]";:B
ASE=2:GOSUB37
21 PRINT"[CRSRD]OCT=[SPC]@[SPC]";:BA
SE=8:GOSUB37
22 PRINT"[CRSRD]DEC=[2SPC]";NR
23 PRINT"[2CRSRD]HEX=[SPC]$[SPC]";:B
ASE=16:GOSUB37
24 ME$="[4SPC]PULSE[SPC]RETURN":GOSU
B26:IFZ$=CHR$(13)THEN2
25 GOTO24
26 FL=1
27 PRINT"[HOM][21CRSRD]";:IFFL=1THEN
PRINTME$
28 IFFL=-1THENPRINT"[21SPC]"
29 FORNL=1TO100:GETZ$:IFZ$<>""THENRE
TURN
30 NEXT:FL=-FL:GOTO27
31 NR=0:X=LEN(DA$)
32 ER=0:FORDI=1TOLEN(DA$):DA=ASC(MID
$(DA$,DI,1))-48:IFDA>9ANDDA<17THENDA
=99
33 IFDA>16THENDA=DA-7
34 IFDA>=BAORDA<0THENPRINT"ERROR":ER
=1
35 NR=NR+DA*BASE↑(X-DI)
36 NEXT:RETURN
37 D$="":DI=0:NZ=NR
38 IFNZ>=BASE↑DITHENDI=DI+1:GOTO38
39 FORX=DI-1TO0STEP-1
40 DA=INT(NZ/(BA↑X)):NZ=NZ-DA*BA↑X
41 IFDA>9THENDA=DA+7
42 D$=CHR$(DA+48):PRINTD$;:NEXT:PRIN
T:PRINT:RETURN
```

Commodore 64 Caracteres Especiales con Easy Script

Cómo generar caracteres especiales con Easy Script

Hemos recibido cartas que nos preguntaban sobre la cuestión de cómo generar caracteres especiales en algunas impresoras con el proceso de textos EASY SCRIPT para Commodore 64.

R. PARDO



C-64

Bien, en las impresoras de matriz de puntos Commodore 1515/1525 y MPS-801, la redefinición de carácter se hace con el comando de EASY SCRIPT ch.

El formato de este comando es el siguiente:

```
*chxx,xx,xx,xx,xx
```

Donde xx son los números que resultan de decodificar los puntos del carácter a generar. (Este tema se toca con mayor profundidad en los manuales de las impresoras 1515/1525 y MPS-801). Por ejemplo:

```
 1 . * * * * * .
 2 . . . . . . .
 4 . * . . . * .
 8 . * * . . * .
16 . * . * . * .
32 . * . . * * .
64 . * . . . * .

    1 2 1 1 1 2
    2 5 3 4 6 5
    8 3 7 5 1 3
```

Luego el comando ch quedará así:

```
*ch128,253,137,145,161,253
```

Para utilizar este carácter especial deberá emplear la combinación de las teclas F1 y $. Esto hará que EASY SCRIPT visualice un símbolo $ en campo de video invertido y, que a la hora de imprimir nuestro texto salga el carácter especial que nosotros hemos definido previamente. ¡Ojo! Recomendamos que la definición de carácter se haga **EN TODOS LOS CASOS** en las primeras líneas de nuestro documento, a poder ser en el lugar donde definimos los parámetros del documento que vamos a escribir.

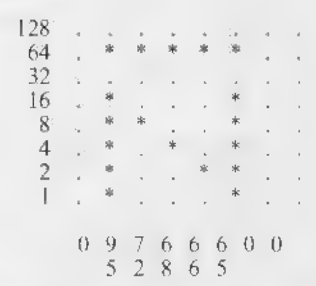
Los usuarios que dispongan de impresoras Commodore 1526 y 4022, 4023, 8023 o MPP-1361 tendrán de utilizar una técnica diferente ya que la manera de definir un carácter especial en estas impresoras no es la misma que en las impresoras que hemos visto anteriormente.

En estas impresoras, hay que seleccionar previamente una dirección secundaria para utilizar cualquiera de sus modos. En el caso que nos ocupa, la dirección secundaria es 5.

¿Cómo enviaremos desde EASY SCRIPT una dirección secundaria? Con el comando sa. El formato de este comando es el siguiente:

```
*saxx,yy,yy,yy,yy,yy,yy
```

Donde xx es el número de dirección secundaria que queremos enviar, e yy son los números que decodificamos de la posición de los puntos del carácter. Vamos a definir el mismo carácter con estas impresoras (en el caso de la impresora 1526):



Pero antes aún hay que ejecutar otra operación.

EASY SCRIPT tiene un método por el cual podemos enviar un código de carácter determinado por nosotros a la impre-

sora. Esto es posible porque EASY SCRIPT permite un máximo de 10 caracteres determinados de este modo. ¿Cómo se hace? Con el comando:

X=ASCII

Donde X es un número del 0 al 9 y ASCII es el valor dado en este código del carácter especial que queramos enviar. En nuestro caso, debemos enviar el carácter ASCCI 254. Por tanto, escribiremos:

*0=254:sa5,0,95,72,68,66,65,0,0

Ahora bien, ¿cómo enviaremos esto? Pues conla combinación de las teclas F1 y 0 (en este caso), debiendo aparecer un 0 en campo de video invertido en la pantalla, y en la impresora el carácter especial que hemos definido. Como hemos visto, es bastante fácil crear caracteres especiales con una impresora de matriz de puntos. Pero ahora pasaremos revista a otro tipo de impresora: las impresoras de **margarita** o de **alta calidad**.

En este tipo de impresoras está disponible caracteres de retroceso (backspace), secuencias de escape (ESC) para distintos modos de gobierno de la impresora, etc.

Vamos a hablar de los dos modelos de impresora de margarita con las cuales trabajamos normalmente en Microelectrónica y Control: **SILVER-REED** y **NEC SPINWRITER.**

Los artículos de esta Revista se imprimen indistintamente en cualquiera de estas dos marcas. Este artículo está redactado con una impresora **SILVER-REED EXP 550** y por tanto, hablaremos en primer lugar de ella.

Esta impresora, en cuanto a códigos de subrayado y repicado, es igual que las impresoras de la marca DIABLO, por tanto algunos de los códigos van a ser iguales para las dos.

Para enviar los caracteres especiales, hemos utilizado el método de enviar caracteres especiales del que dispone EASY SCRIPT. Veamos qué caracteres empleamos:

*0=125:nb c con cedilla (ç)
*1=92:nb eñe mayúscula (Ñ)
*2=124:nb eñe minúscula (ñ)
*3=64:nb acento (′)
*4=126:nb símbolo 1/2 (½)
*5=62:nb símbolo pesetas (Pts)
*6=60:nb símbolo de primero (º)
*7=123:nb símbolo de primera (ª)
*8=93:nb abrir interrogación (¿)
*9=91:nb abrir admiración (¡)

Como se puede ver, todos estos códigos permiten hacer un proceso de texto de calidad. Cualquiera de estos signos se puede utilizar pulsando F1 y el número correspondiente.

Ahora bien, si probamos el acento más de uno se encontrará con la sorpresa de que el acento ocupa un espacio y la vocal sobre la cual iba destinada la tilde ocupa el espacio de al lado. Esto quiere decir que nos falta algo.

Estas impresoras disponen de caracteres de **retroceso** o **backspace.** ¿Cómo enviaremos un carácter de este tipo con EASY SCRIPT? Con la combinación de las teclas F1 y flecha a la izquierda se envía un carácter de backspace con la ventaja que este carácter que retrocede el carro de la impresora se toma en cuenta por parte de EASY SCRIPT a la hora de justificar márgenes.

Nosotros, cuando tenemos de enviar un acento, enviamos primero F1/3, saliendo en la pantalla un 3 en campo de video invertido, y luego F1/flecha izquierda, saliendo una B mayúscula en campo de video invertido. Se puede invertir el orden de envío con idénticos resultados.

Aún hay otro tema, pues en castellano existen palabras que tienen diéresis, como cigüeña. ¿Cómo enviaremos el carácter¨?

Por medio de la combinación de la función ESC de la impresora. Para enviar este carácter desde EASY SCRIPT, tendremos de pulsar F1/flecha arriba, apareciendo una E mayúscula en campo invertido en la pantalla, poniendo a continuación una Y mayúscula (esto no es que me lo haya inventado yo, sino que según el manual de esta impresora es la manera correcta de sacar la diéresis).

¿Cómo lo utilizaríamos en el caso de la palabra cigüeña?

Escribiríamos las letras c, i y g. Ahora F1/flecha arriba y luego Y mayúscula. Luego F1/flecha izquierda, u, e, ñ, a. ¿Qué fácil, no?

Hablaremos de otra marca de impresora: la **NEC SPINWRITER.**

Estas impresoras no tienen los mismos códigos de carácter para generar los mismos caracteres que la impresora de la que hemos hablado anteriormente.

Por ejemplo, para estas impresoras hay que enviar previamente el comando de EASY SCRIPT lf con un valor de 1 ya que vienen preparadas para ejecutar Carriage Return sin Line Feed, o si no nos imprimiría siempre en la misma línea ya que no avanzaría el papel.

Para generar el acento en estas impresoras, enviaremos el caracter ' (SHIFT y 7) seguido de backspace (F1/flecha izquierda). Por ejemplo para escribir condición:

condici'Bon

En cuanto a la eñe, no tenemos un carácter especial como en el caso de SILVER REED sino que tendremos de definir un carácter especial: la tilde (esa línea de encima de la eñe). Por tanto:

*0=126

nos enviará este carácter. Luego para generar la eñe tendremos que enviar primero la tilde, luego backspace y luego ene. Por ejemplo así escribiríamos año:

a0Bno

Así pues, hemos visto lo fácil que es generar caracteres especiales con EASY SCRIPT y cualquier impresora. Creo que con este artículo podrá cualquier persona crear y/o enviar cualquier tipo de carácter desde este proceso de texto.

METEDURAS DE PATA

En el listado de "tiro al blanco" para el C-64 del número uno, se deslizaron varios errores en las líneas 750, 920, 940 y 970, deben ser:

```
750 POKE NI + DF, WHITE
920 PRINT" [YEL] [40SHIFT Q]"
940 PRINT" [YEL] [40SHIFT Q]
    [3CRSRD]"
970 IF HITS  HS THEN HS = HI:
    PRINTTAB(8)"[GRN][CRSRD]
    NEW [SPC] HIGH [SPC]
    SCORE! [BLK]"
```

En el número 2 de la revista se han colado los siguientes:

Pág. 11, en el cuadro donde dice C-24 debe decir C-64.

Pág. 18, donde dice "recomendado para C-64 y C-700" debe decir "para C-700".

Pág. 19, donde dice "fotos cedidas por DENO" debe decir "por DEMO".

Pág. 44, donde dice "$0304 783 almacenamiento temporal...", debe decir "$030F 783..."

Pág. 45, donde dice "1º Cuando se utilizan $KRAM..." debe decir "1º Cuando se utilizan 8 KRAM..."

Pág. 48, En el último truco la línea 10000 debe ser:

```
10000 POKE 781, FILA: POKE 782,
COLUMNA: POKE783,0: SYS 65520:
RETURN
```

ATENCION — JOSE Mª MACI = Encantados de hablar contigo por teléfono **pero** Tiro al Blanco **no** nos da error en la línea 850.

En la colaboración de Ernesto Sánchez UTILIZANDO EL PORT DEL USUARIO publicado en el número de febrero se colaron un par de espontáneas "colaboraciones" de nuestro BUG particular: En la página 29, columna derecha, párrafos 3 y 4 se dice PB0, PB3 y PB7. Pues bien, debe decir: PB0, PB3 y PB6.

En el número 2 no publicamos las fichas de descripción alfabética de los mnemónicos del 6502/6510 por tardanza de correos. Ya se han tomado medidas para que no vuelva a suceder.

CORRECCIONES DEL MANUAL

Manual del usuario de la Unidad de Disco VIC 1541

Página	Error
14...	Programa soporte del DOS. No menciona nada de dos de los comandos del C-64 (no valen en el VIC), son: % NOMBRE = LOAD" NOMBRE",8,1 % NOMBRE= LOAD" NOMBRE",8+RUN
64...	Formato para el comando "position". Todas las veces que aparece en el ejemplo la abreviatura re9 deben sustituirse por reg. (registro).
65...	Primer listado. La línea 40 debe ser: 40 PRINT#1, "p" CHR$(3)CHR$(1) CHR$(0)CHR$(1)
65...	Segundo listado. La línea 80 debe ser: 80 RESTORE: DATA 1, FIRST NAME, ... etc ...
67...	Método de hardware. Las nuevas unidades de disco tienen otra placa de circuito impreso diferente a las antiguas, y el punto 7 del cambio es casi imposible por no estar los puentes en la misma posición. Los nuevos puentes se encuentran en el centro de la placa; en el próximo número publicaremos un croquis sobre su ubicación.
35...	En el programa para leer el directorio faltan las líneas 500 a 550, que deberían ser las siguientes: 500 B$="" 510 FOR L=0 TO J 520 GET#1,A$ 530 IF A$ CHR$(96) THEN IF A$ CHR$(160) THEN B$=B$+A$ 540 NEXT 550 RETURN

DESCRIPCIÓN ALFABÉTICA DE LOS MNEMÓNICOS DEL 6502/6510 (IV)

BNE

BNE

Salta si resultado distinto de cero

Operación: Salta si Z = 0

(Ref.: 4.1.1.6)

N Z C I D V
— — — — — —

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Relativo	BNE Oper.	D0	2	2*

* Suma 1 si se salta a la misma página.
* Suma 2 si se salta a otra página.

BRK

BRK

Fuerza una interrupción

Operación: PC + 2 P

(Ref.: 9.11)

N Z C I D V
— — — 1 — —

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	BRK	00	1	7

* El comando BRK no puede eliminarse activando I.

BPL

BPL

Salta si el resultado es positivo

Operación: Salta si N = 0

(Ref.: 4.1.1.2)

N Z C I D V
— — — — — —

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Relativo	BLP Oper.	10	2	2*

* Suma 1 si se salta a la misma página.
* Suma 2 si se salta a otra página.

BVC

BVC

Salta si no hay desbordamiento

Operación: Salta si V = 0

(Ref.: 4.1.1.8)

N Z C I D V
— — — — — —

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Relativo	BVC Oper.	50	2	2*

* Suma 1 si se salta a la misma página.
* Suma 2 si se salta a otra página.